Estruturas Multidimensionais

Profa. Barbara Quintela
Prof. Jose J. Camata
Prof. Marcelo Caniato
barbara@ice.ufjf.br
camata@ice.ufjf.br
marcelo.caniato@ice.ufjf.br

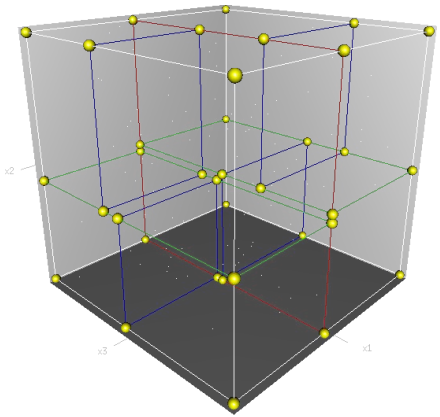
Introdução

- Determinadas aplicações necessitam de estruturas de dados que permitem uma consulta eficiente envolvendo chaves multidimensionais
- Exemplos de Aplicações:
  - Processamento de imagem - 2 dimensões
  - Projeto assistido por Computador - CAD - 2 ou 3 dimensões
  - Astronomia (simulação de galáxias) - 3 dimensões
  - Compressão de dados com perdas - Animação - 3 a 4 dimensões
  - Bancos de dados geográficos - 2 ou 3 dimensões
- Necessidade de usar estruturas de dados especiais para ser processados, recuperados e removidos eficientemente.

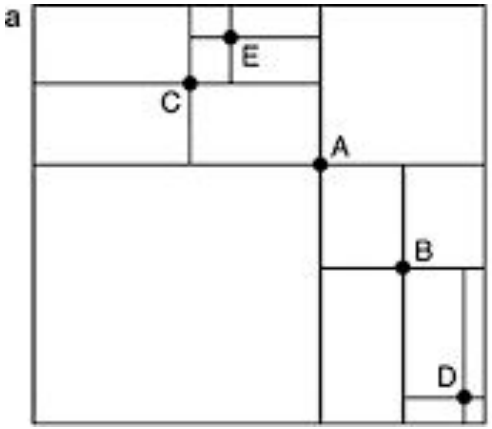
Estruturas Multidimensionais

- Alguns dos métodos de acesso multidimensionais mais conhecidos:

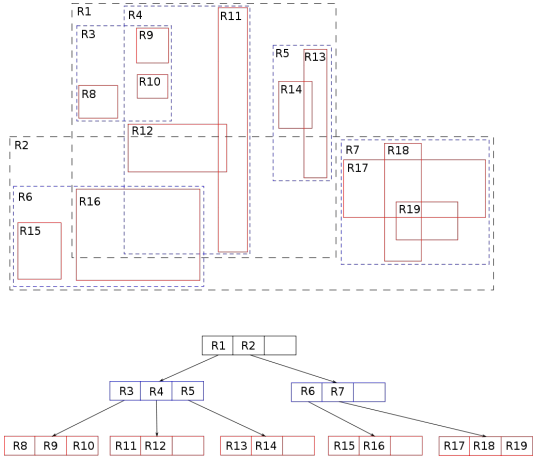
K-d Tree



Quadtree



R-Tree



K-D Tree

- Árvore de busca binária de d dimensões que se caracteriza por testar, em cada nível percorrido, uma dimensão k da chave;
- Chave está associada a um discriminador.
  - *Discriminador indica qual dimensão da chave divide os nós restantes nas duas subárvores esquerda e direita.*

Representação de nó em uma K-d Tree

Chave: [0,.., d-1]	info
discriminador	
Esquerda	Direita

Exemplo:

- Pontos no espaço bidimensional (x,y)
- Chave: [x,y]

K-D Tree

- Nome originalmente 3-d tree, 4-d tree etc
- k é o # de dimensões
- hoje se diz K-d Tree de dimensão d

Ideia: cada nível da árvore compara com uma dimensão

K-D Tree

Considerando d=2, cada nó tem um ponto **P = (x,y)**

Para encontrar **(x',y')** compara a coordenada da dimensão de corte

Ex: Se a dimensão de corte é a dimensão x:

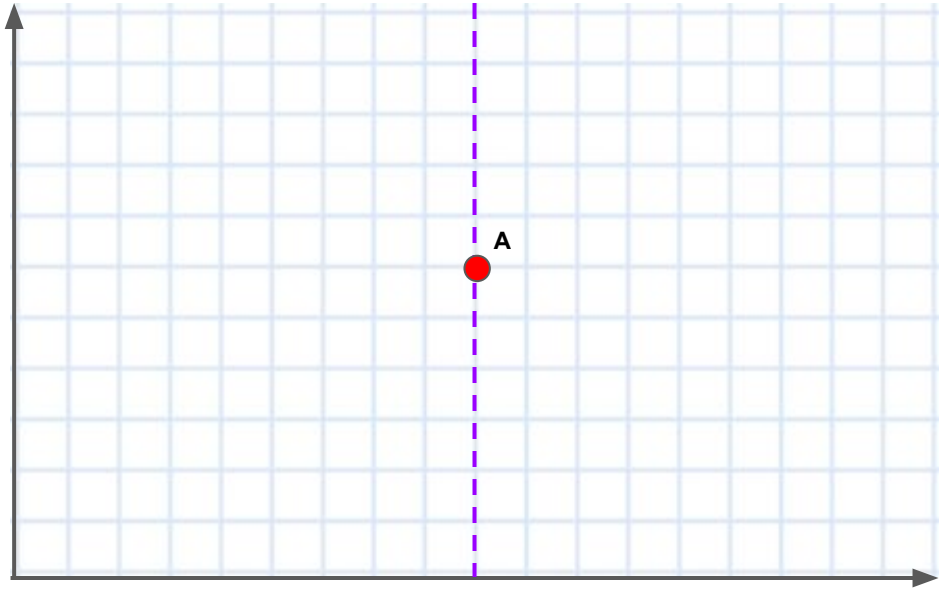
então a pergunta é : **x'** < **x**?

Visualizador:
http://lti.cs.vt.edu/OpenDSA/AV/Development/kd-treeAV.html

Kd Tree - Exemplo

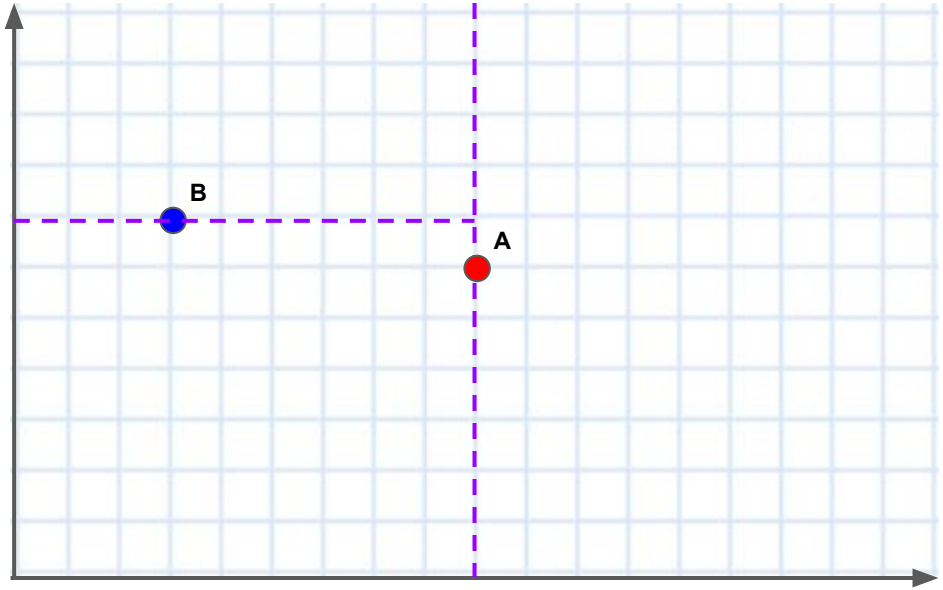
Inserções:

Kd Tree - Exemplo



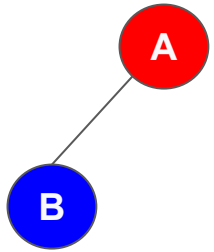
Inserções:
A (9,6)

Kd Tree - Exemplo

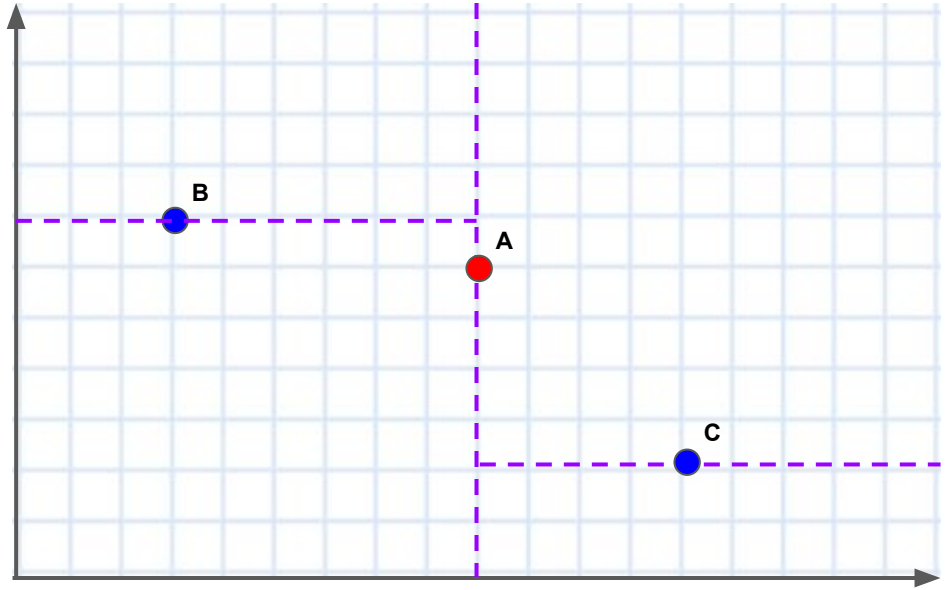


Inserções:

A (9,6), B(3,7)

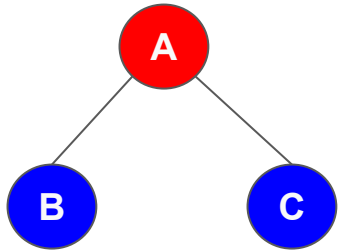


Kd Tree - Exemplo

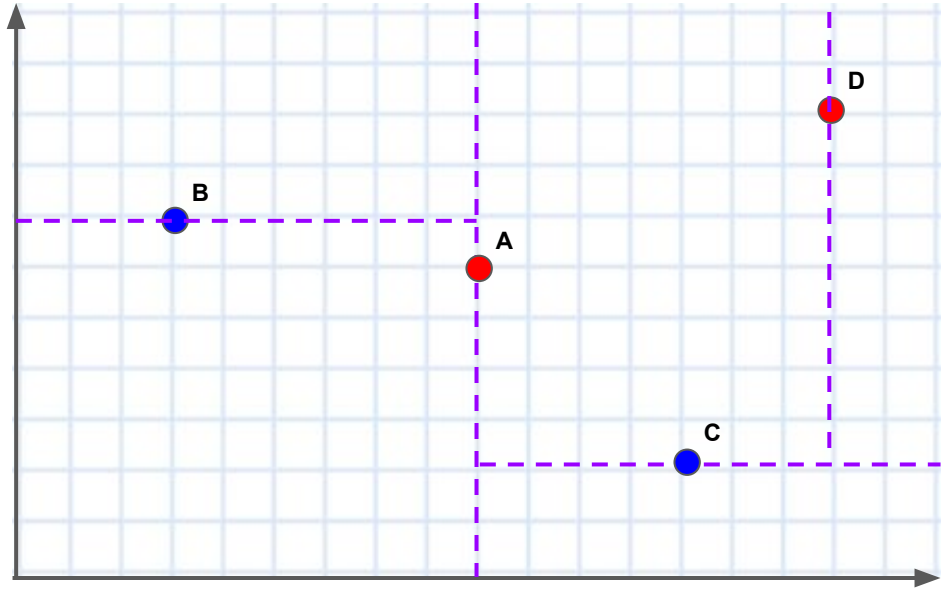


Inserções:

A (9,6), B(3,7), C(13,2)

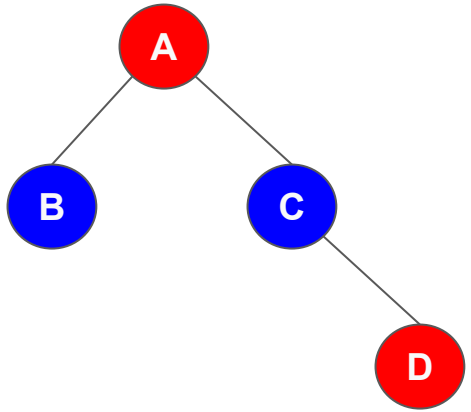


Kd Tree - Exemplo

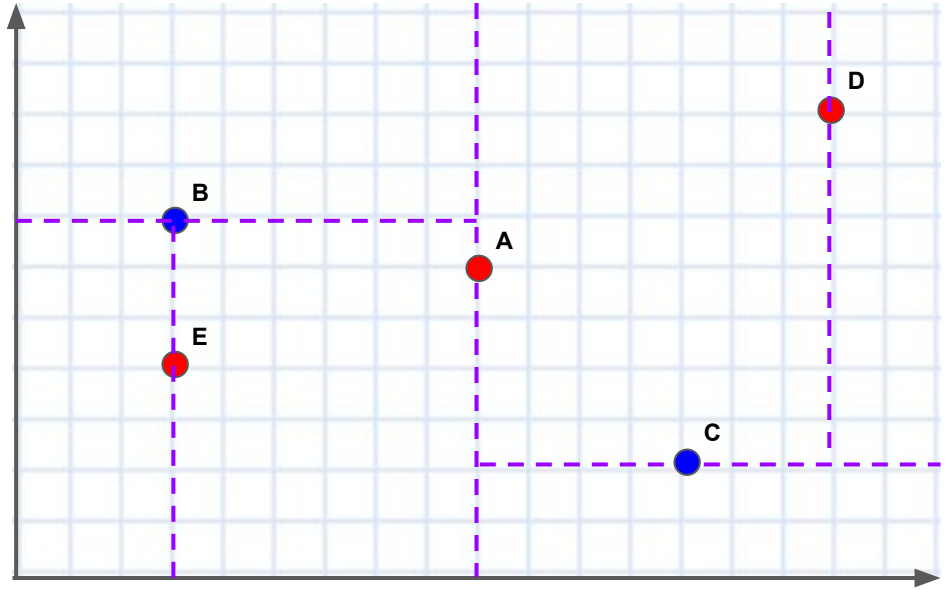


Inserções:

A (9,6), B(3,7), C(13,2),D(16,9)

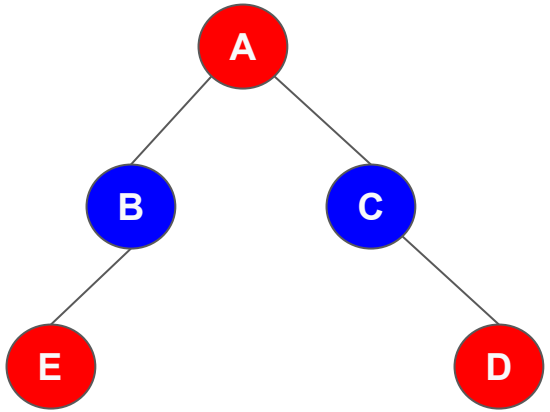


Kd Tree - Exemplo

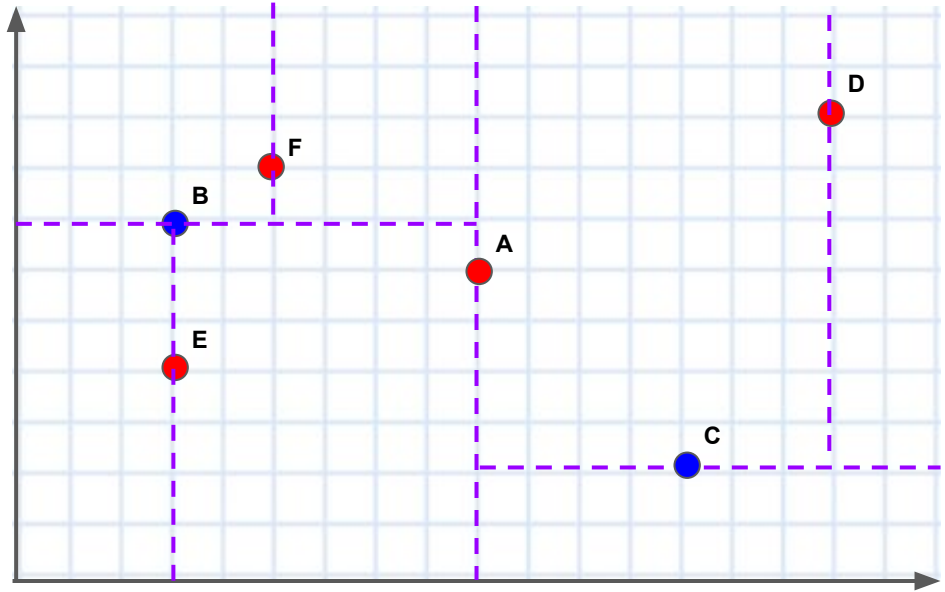


Inserções:

A (9,6), B(3,7), C(13,2),D(16,9), E(3,4)

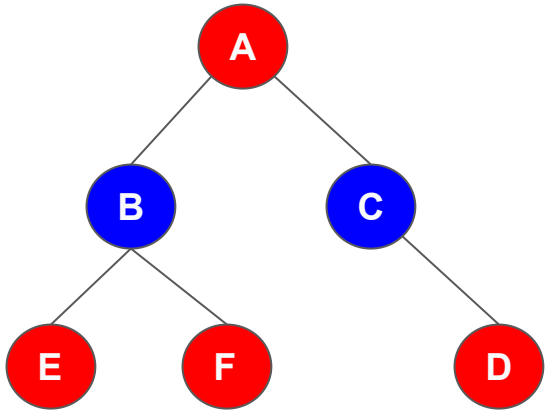


Kd Tree - Exemplo

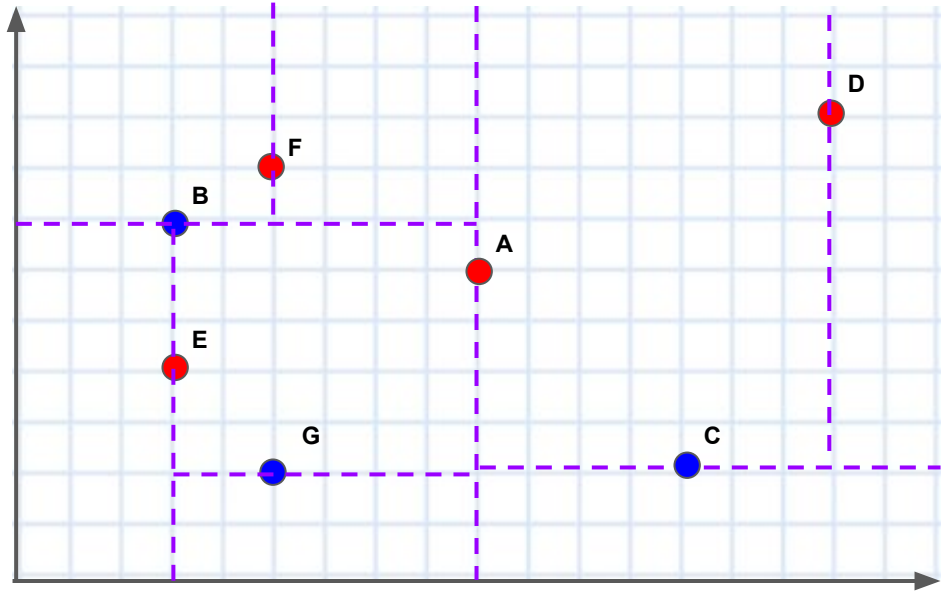


Inserções:

A(9,6), B(3,7), C(13,2),D(16,9), E(3,4), F(5, 8)

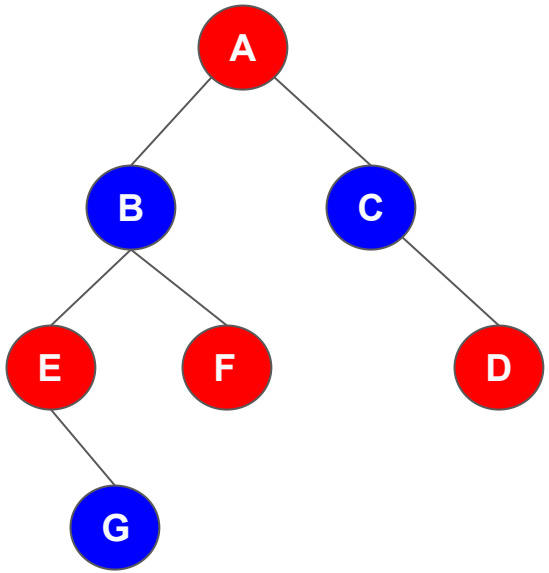


Kd Tree - Exemplo



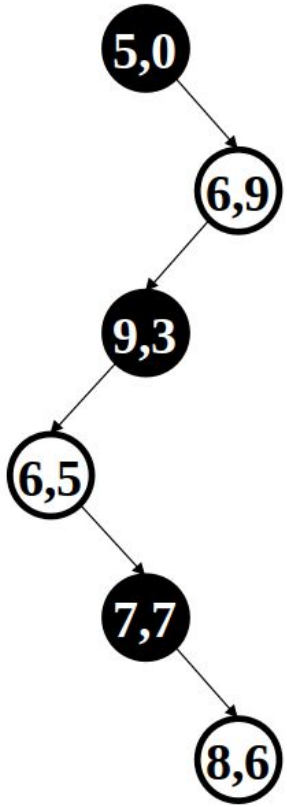
Inserções:

A(9,6), B(3,7), C(13,2),D(16,9), E(3,4), F(5, 8), G(5,2)



K-d Tree

Pode ser ineficiente!
A ordem de inserção pode gerar uma árvore desbalanceada.



K-D Tree (balanceada)

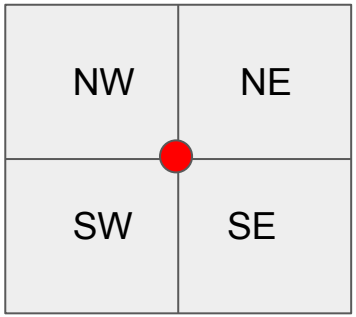
- A árvore é construída por uma função recursiva

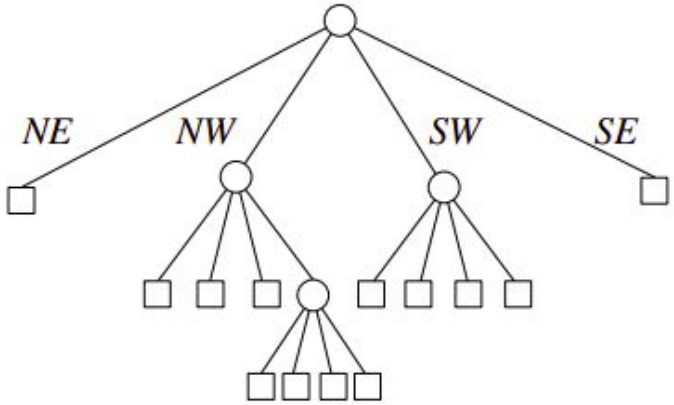
Algoritmo ConstroiArvKD(P,prof) // P é conjunto de pontos e prof é a profundidade

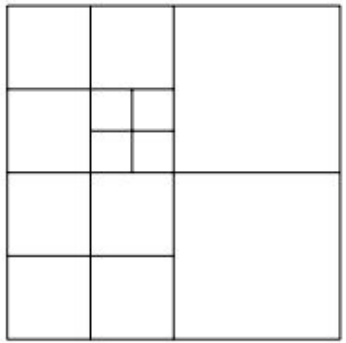
```
1. SE P contém apenas um ponto ENTÃO
2.    retorna uma folha contendo este ponto
3. SENÃO SE prof é plano ENTÃO
4.    divida P em dois subconjuntos com uma linha vertical pela coordenada x
mediana dos pontos em P. P1 é o conjunto de pontos da esquerda e P2 o conjunto
de pontos da direita. Pontos exatamente na linha pertencem a P1
5.  SENÃO
6.    divida P em dois subconjuntos com uma linha horizontal pela coordenada y
mediana dos pontos em P. P1 é o conjunto de pontos acima da linha e P2 o
conjunto de pontos abaixo da linha. Pontos exatamente na linha pertencem a P1.
7. Vright <- ConstroiArvKD(P1,prof+1).
8. Vleft <- ConstroiArvKD(P2,prof+1).
9. Crie um nodo V com Vright e Vleft juntamente com seus filhos direito e
esquerdo, respectivamente.
10. retorne V.
```

Quadtrees

- Uma quadtree é uma árvore de busca enraizada na qual cada nó interno tem quatro filhos.
- Os filhos da raiz são rotulados NE, NW, SW e SE para indicar a qual quadrante eles correspondem;







Quadtrees

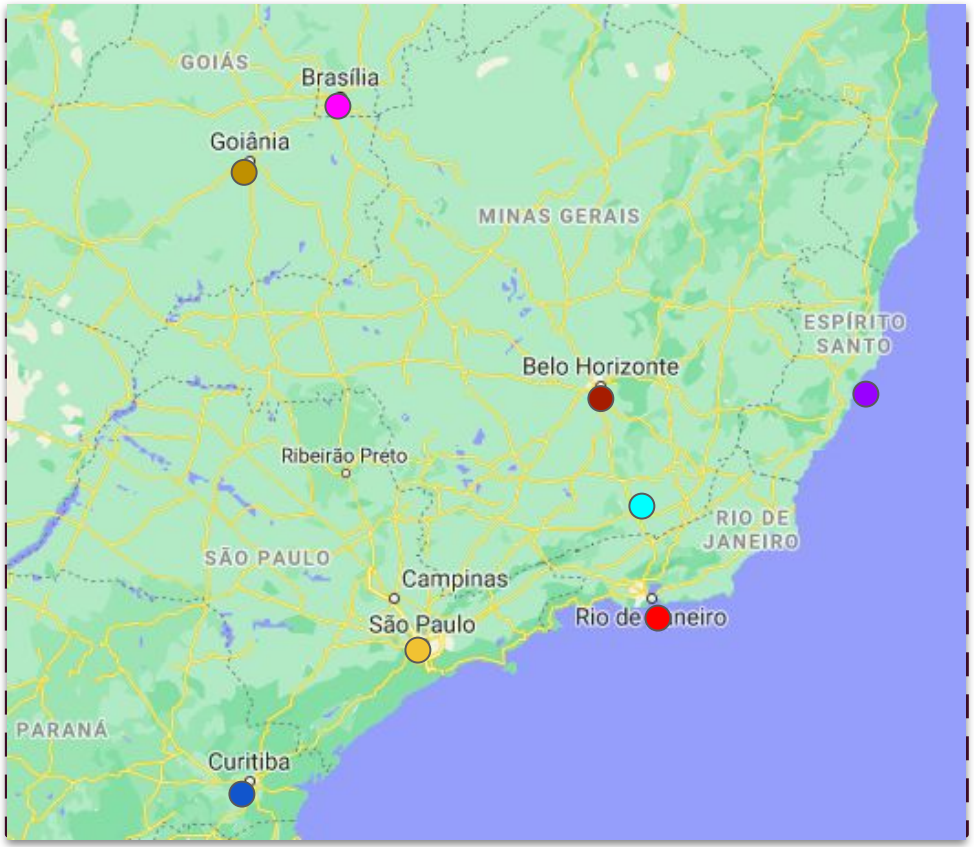
- Quadtrees podem ser usadas para armazenar diferentes tipos de dados.
  - **Várias aplicações**: computação gráfica, jogos, filmes, visão computacional, CAD, mapas (Google Maps, Google Earth), realidade virtual…
- Divisões do espaço devem ser realizadas segundo determinadas características ou regras!
- Vamos descrever a variante que armazena um conjunto de pontos no plano.

Point Quadtrees

- Proposta por Finkel e Benkley [1974]
- É implementada como uma generalização multidimensional de uma árvore binária de busca.
- Em duas dimensões cada ponto de dados é representado por um nó da quadtree:
  - CAMPO: contém informações descritivas sobre o nó, por exemplo, nome da cidade.
  - X e Y: Coordenadas do ponto
  - NE,NW,SW,SE: ponteiros para os 4 quadrantes filhos

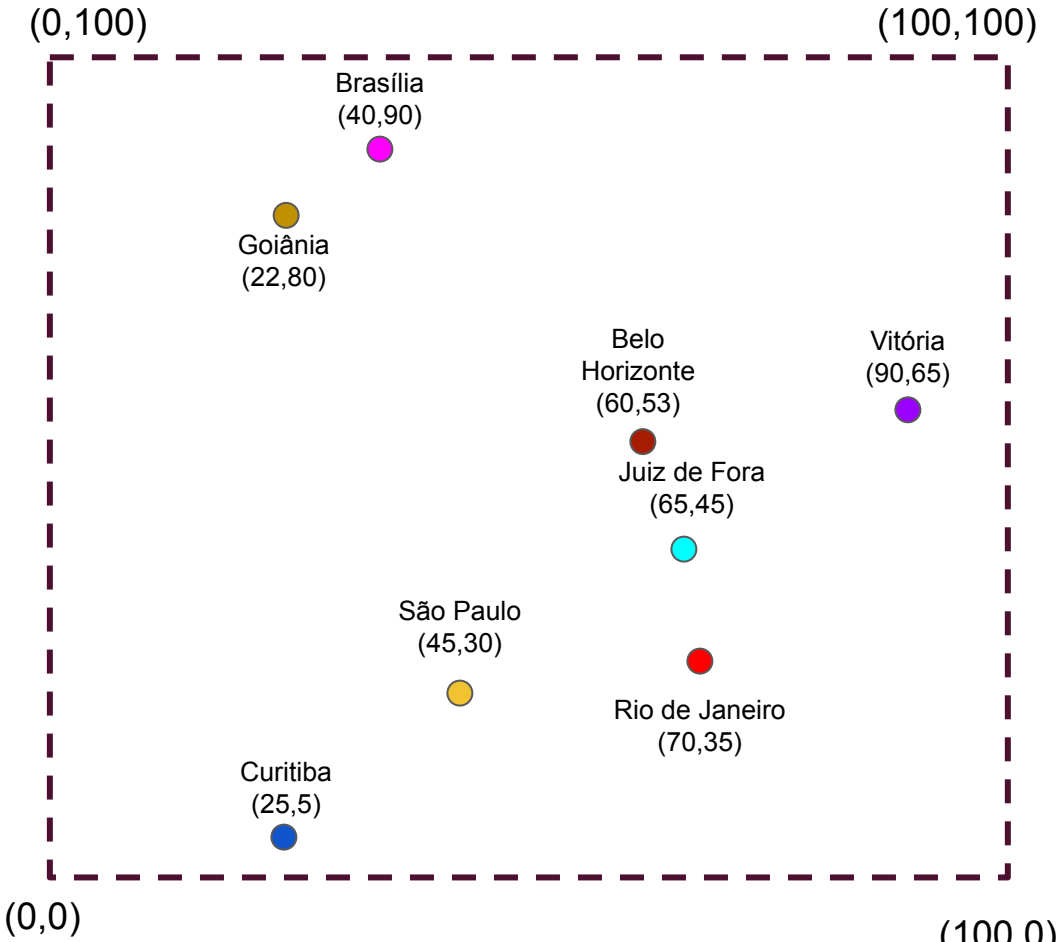
<table>
<tr><td colspan="4">CAMPO</td></tr>
<tr><td colspan="2">COORDX</td><td colspan="2">COORDY</td></tr>
<tr><td>NE</td><td>NW</td><td>SW</td><td>SE</td></tr>
</table>

Exemplo
Localização de Cidades em mapas



Quadtree - Exemplo

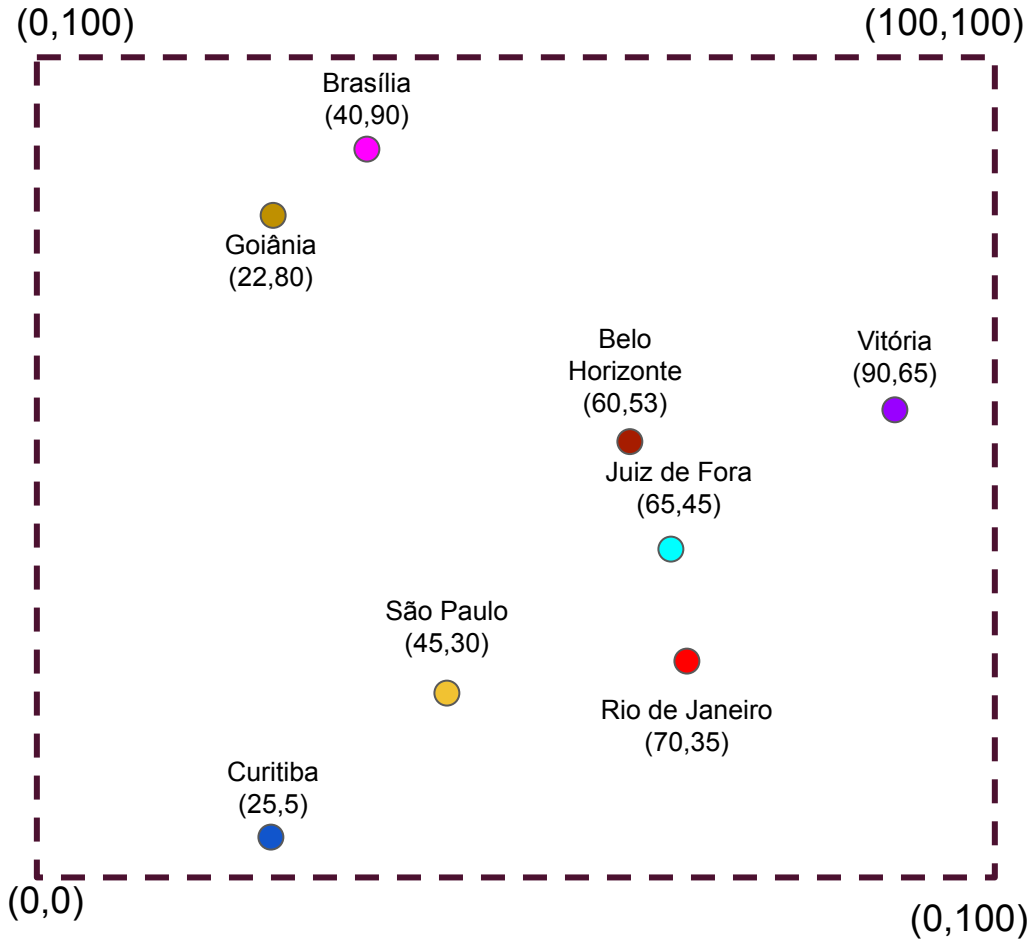
1. Definir uma região de contorno que limite todos os pontos.
2. Pontos serão inseridos na seguinte ordem:
   a. São Paulo
   b. Juiz de Fora
   c. Goiânia
   d. Belo Horizonte
   e. Vitória
   f. Curitiba
   g. Brasília
   h. Rio de Janeiro



Quadtree - Exemplo

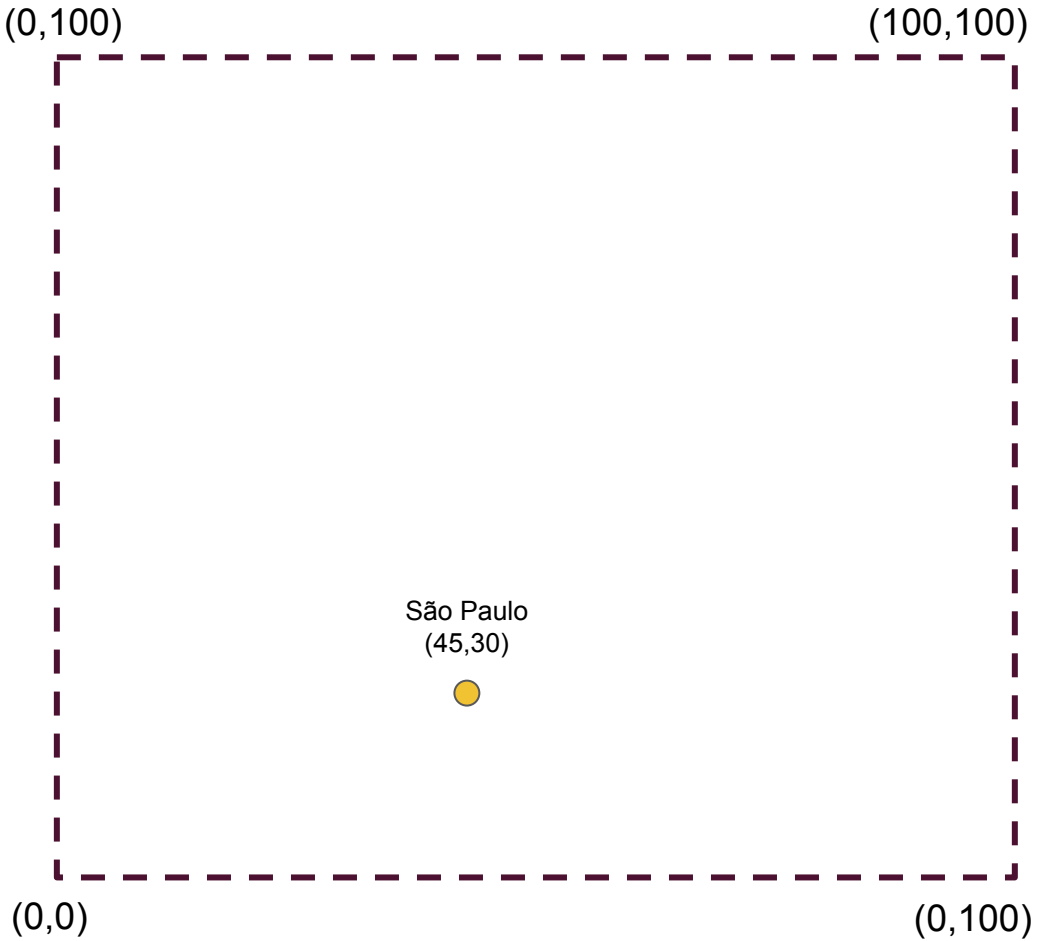
INSERÇÃO

1. Registros são inseridos de forma semelhante à árvore binária de busca.
2. A posição desejada é buscada de acordo com as coordenadas X e Y.
3. Em cada nó uma comparação é feita e a subárvore apropriada (NE, NW, SW ou SE) é escolhida.
4. Quando chega-se na base da árvore (filho nulo), foi encontrada a posição desejada para inserir o registro.



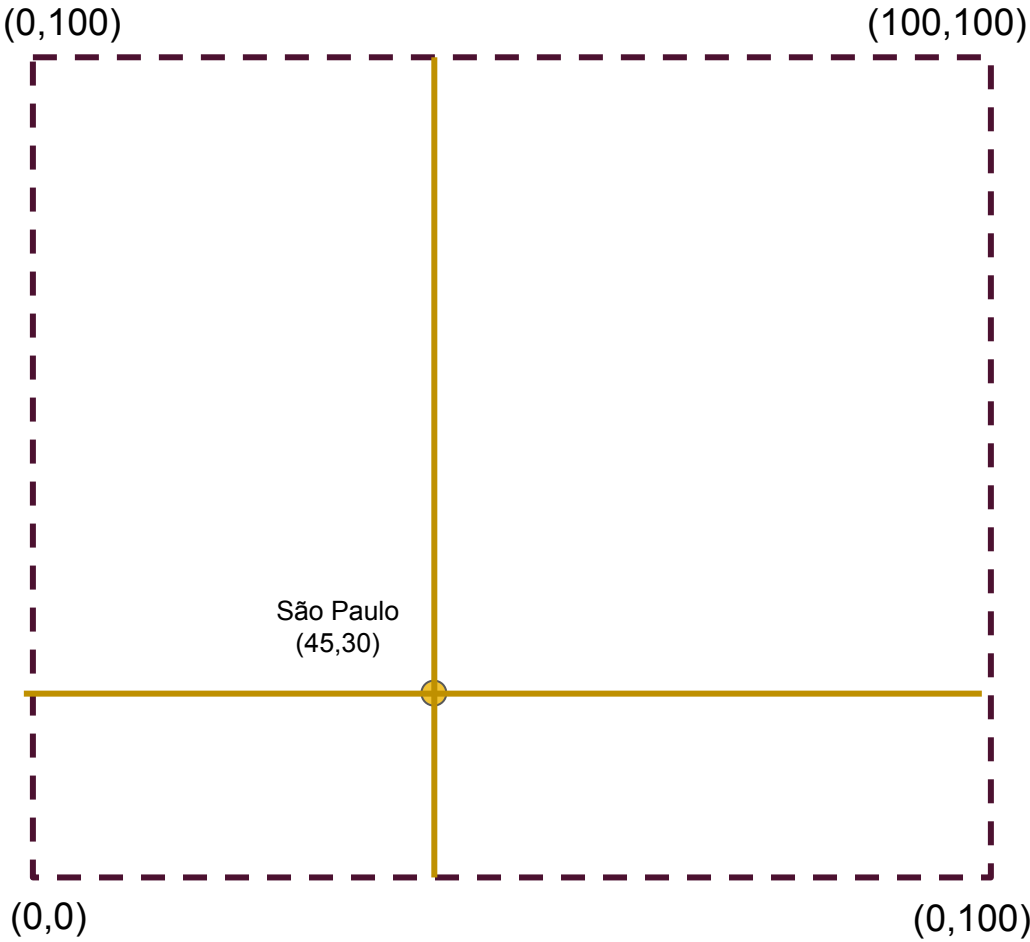
Quadtree - Exemplo

Inserção: São Paulo



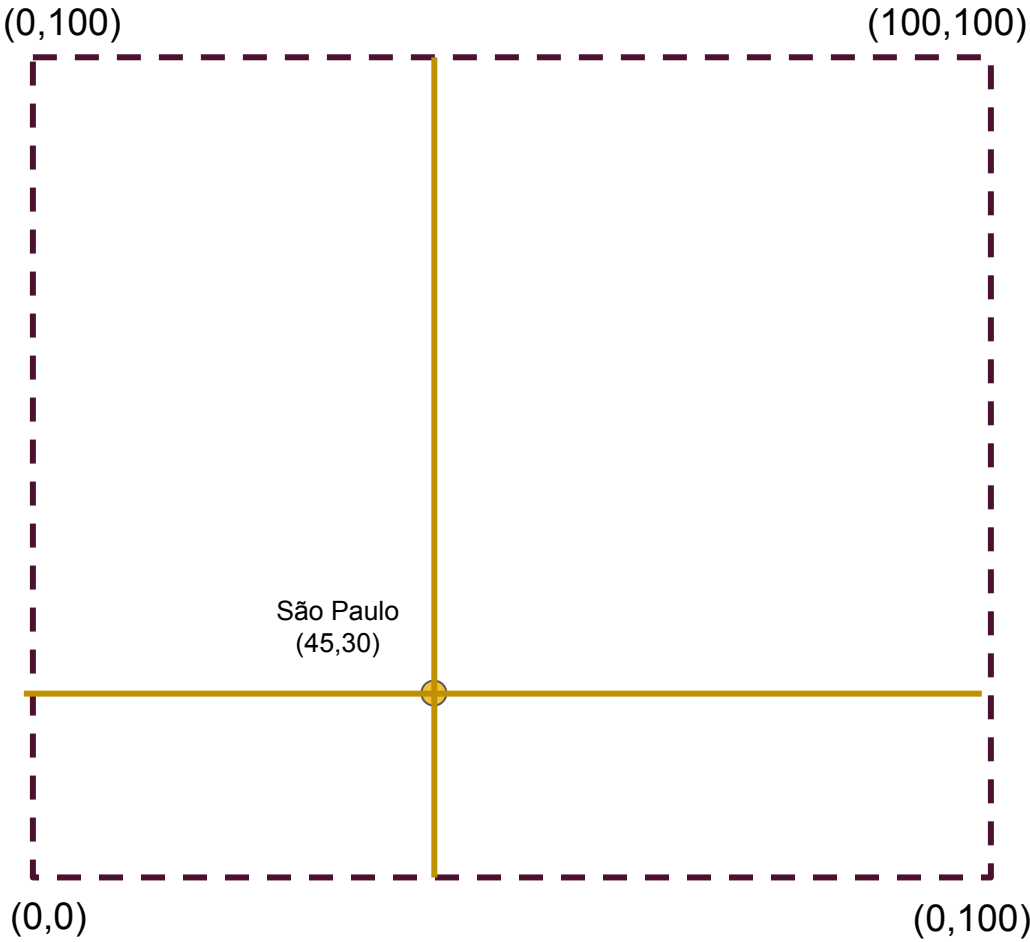
Quadtree - Exemplo

Inserção: São Paulo



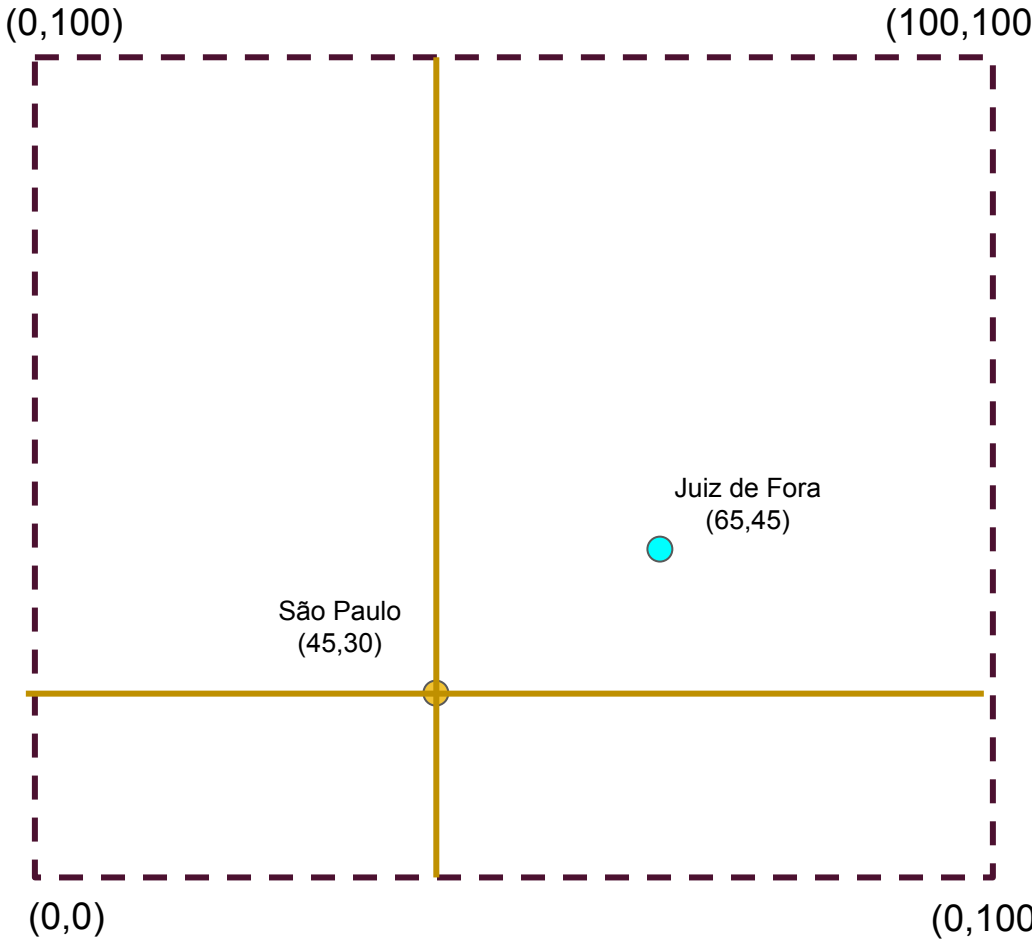
Quadtree - Exemplo

Inserção: Juiz de Fora



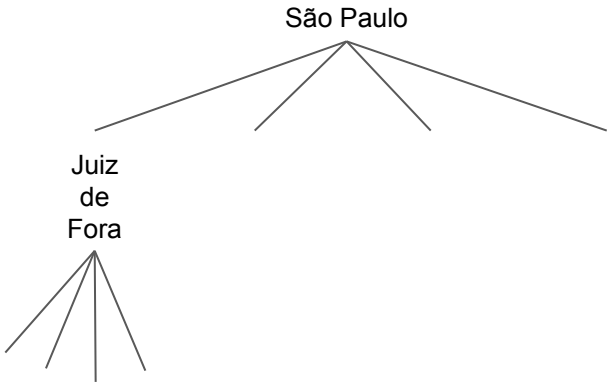
Quadtree - Exemplo

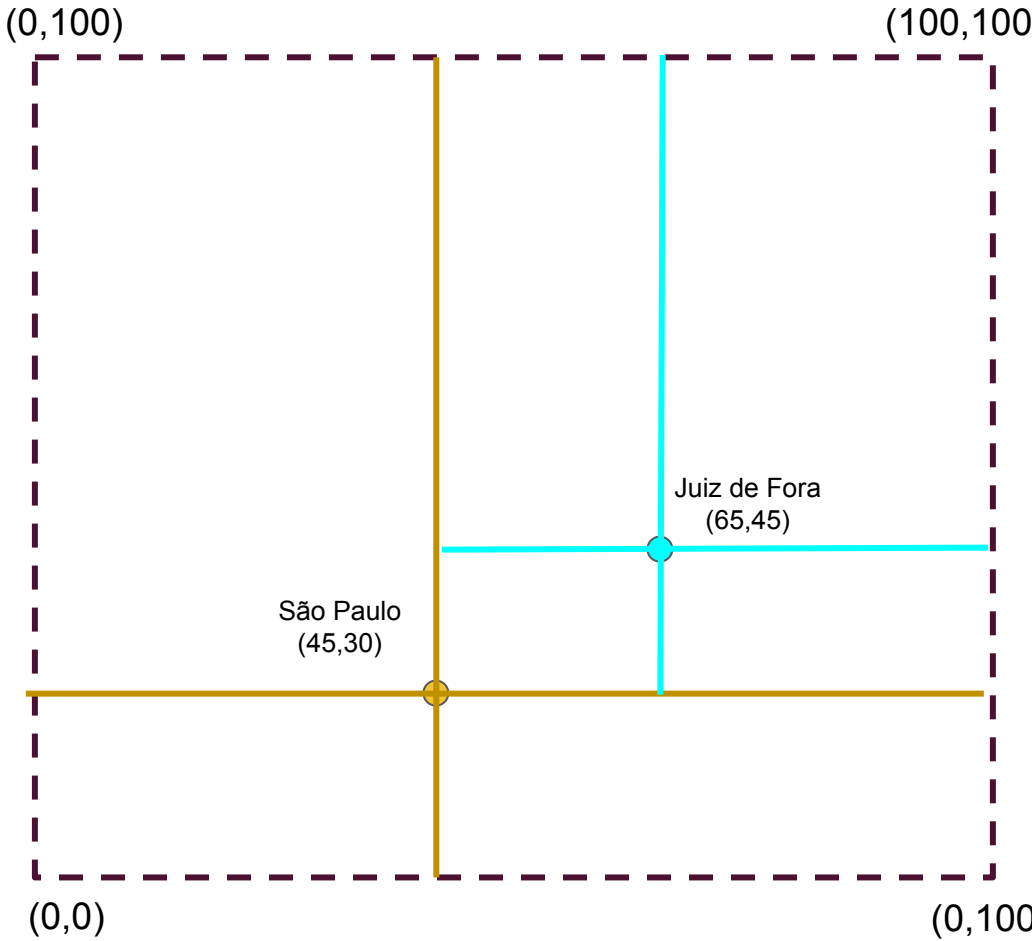
Inserção: Juiz de Fora



Quadtree - Exemplo

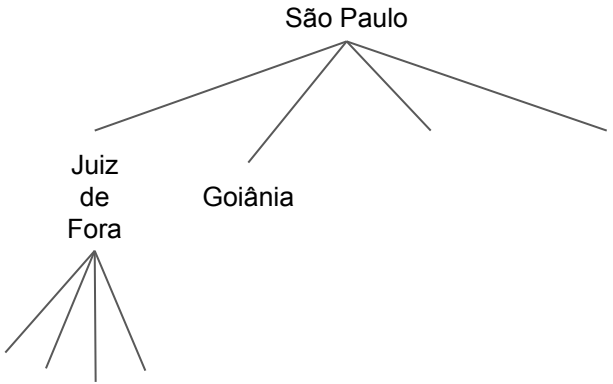
Inserção: Juiz de Fora

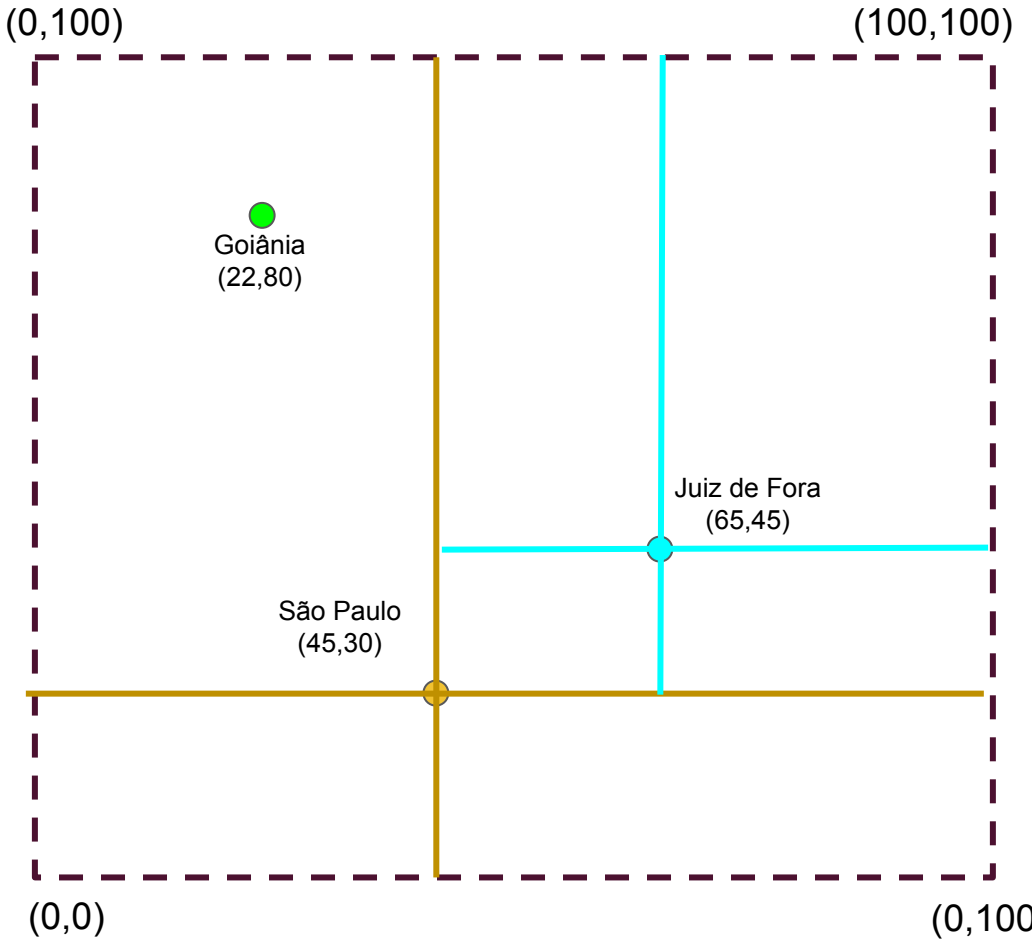




Quadtree - Exemplo

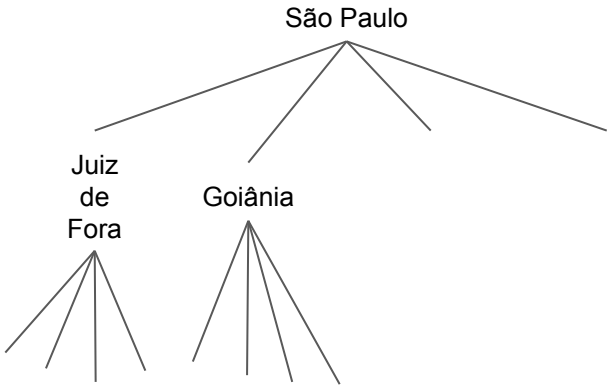
Inserção: Goiânia

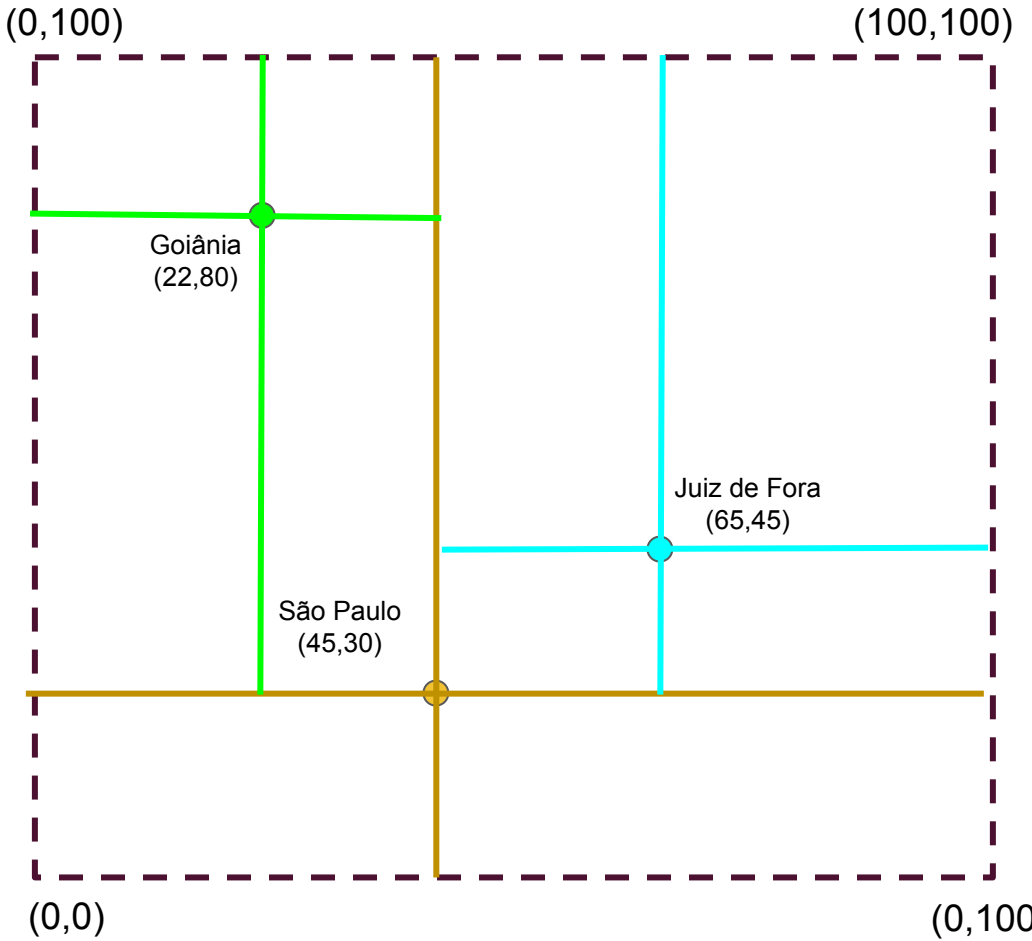




Quadtree - Exemplo

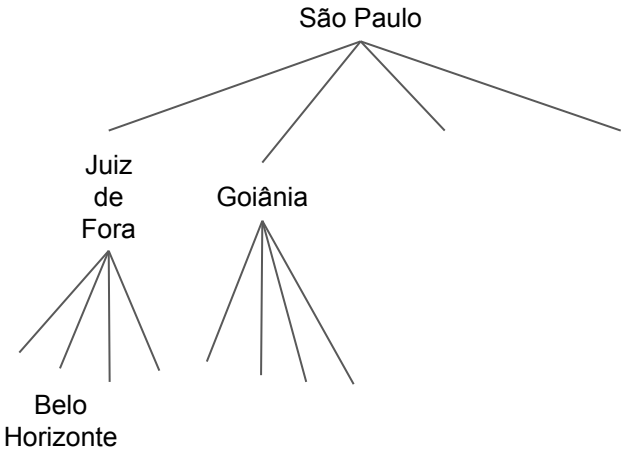
Inserção: Goiânia

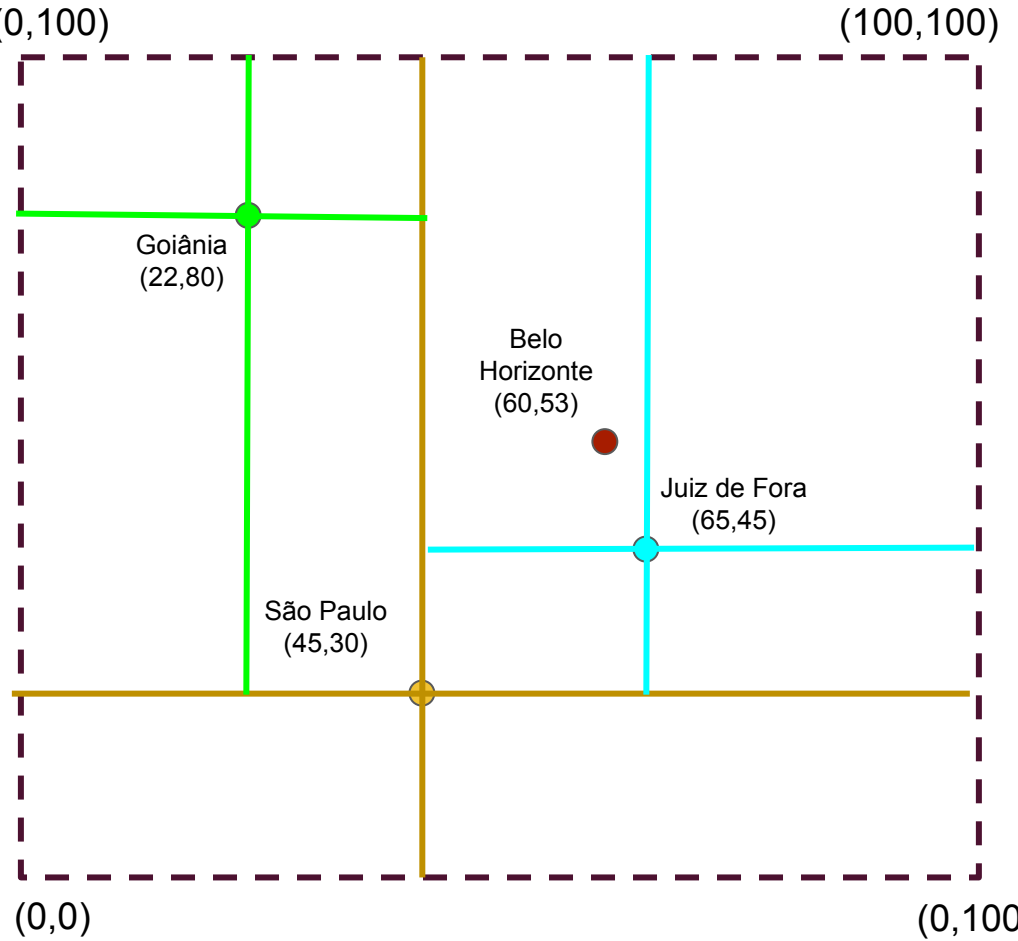




Quadtree - Exemplo

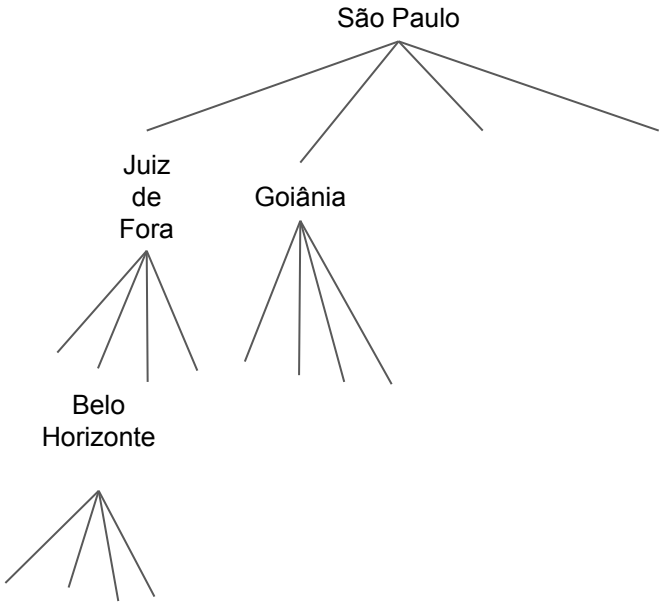
Inserção: Belo Horizonte

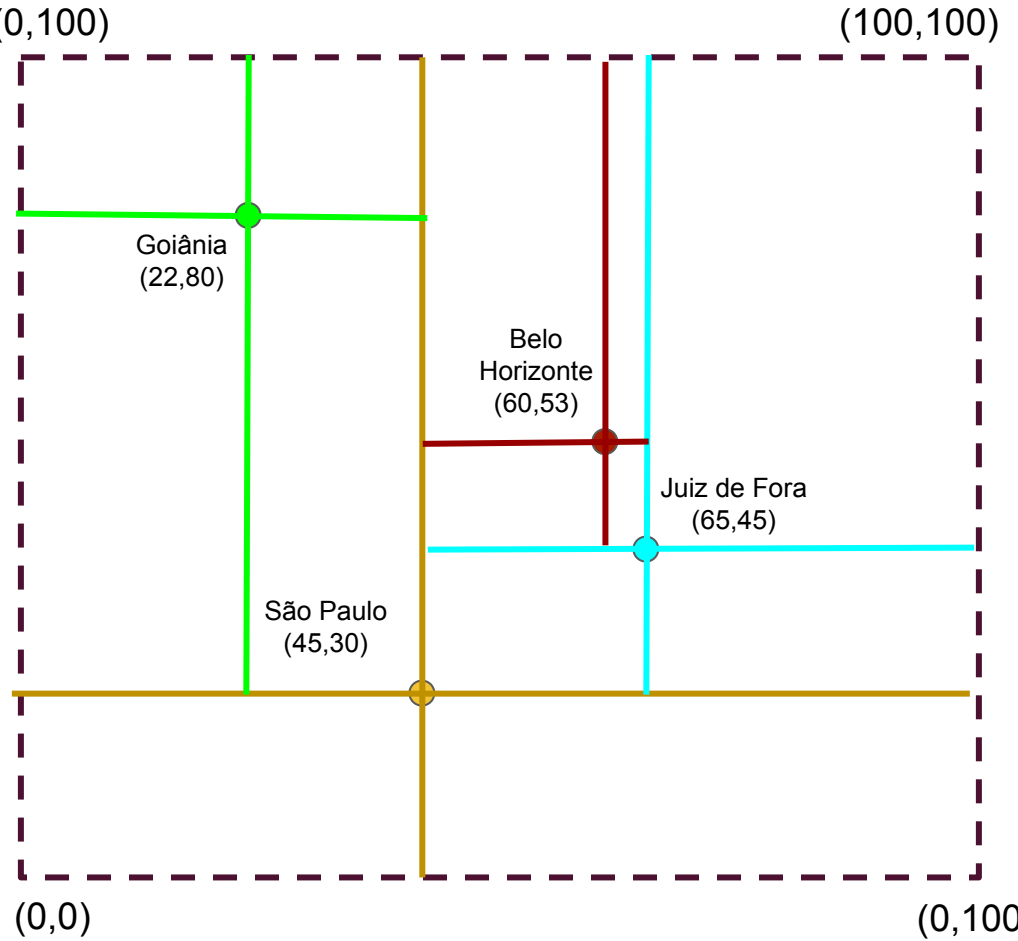




Quadtree - Exemplo

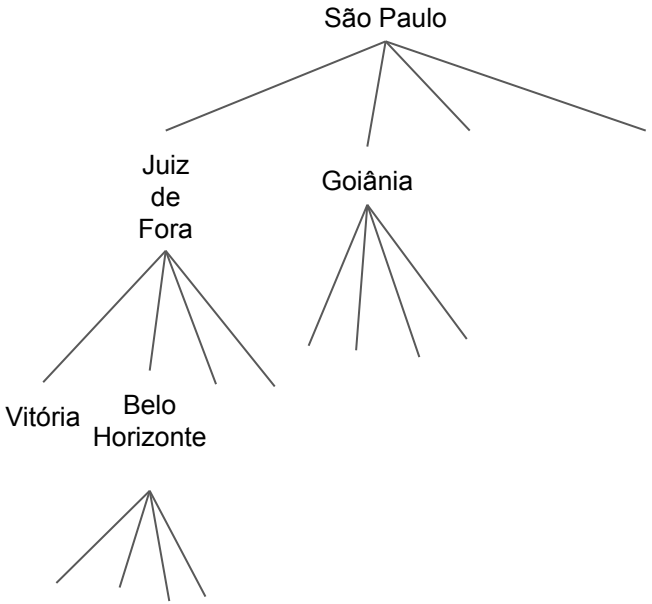
Inserção: Belo Horizonte

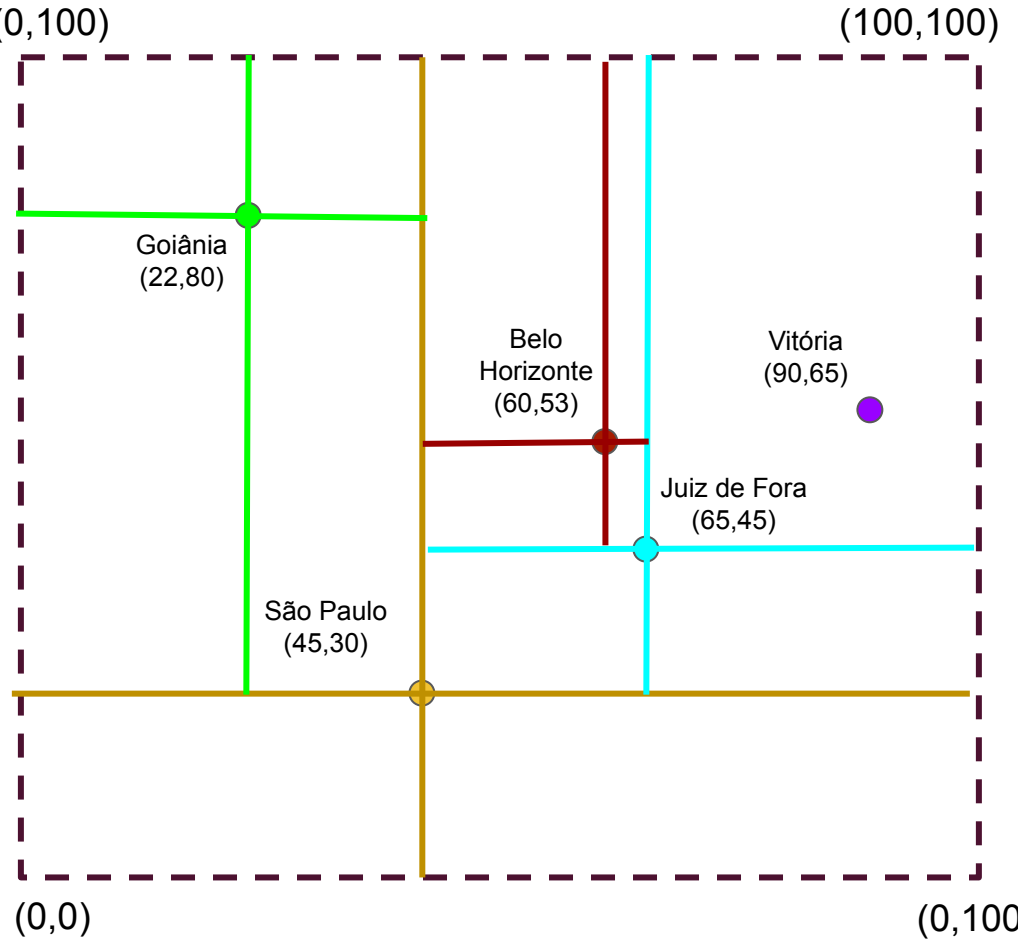




Quadtree - Exemplo

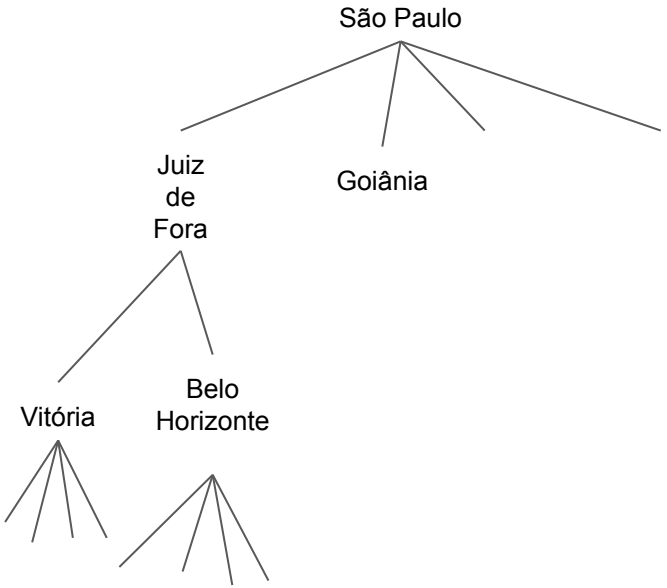
Inserção: Vitória

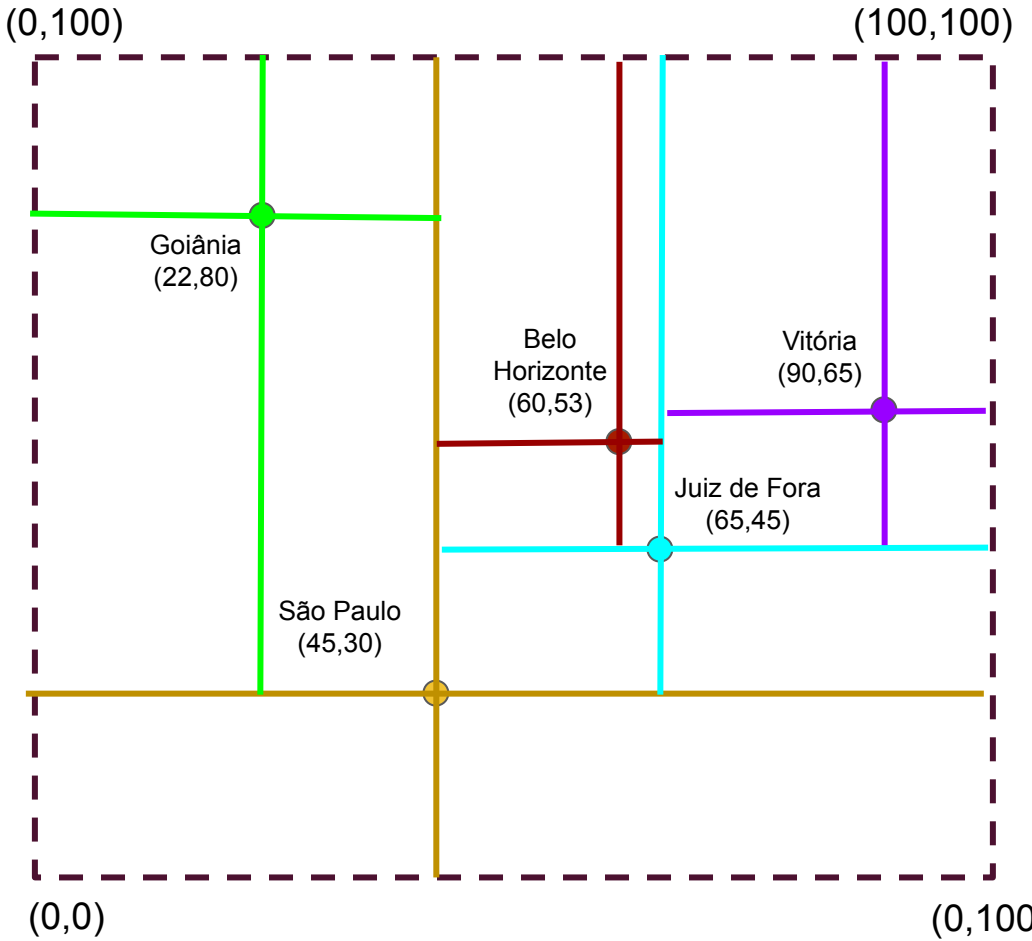




Quadtree - Exemplo

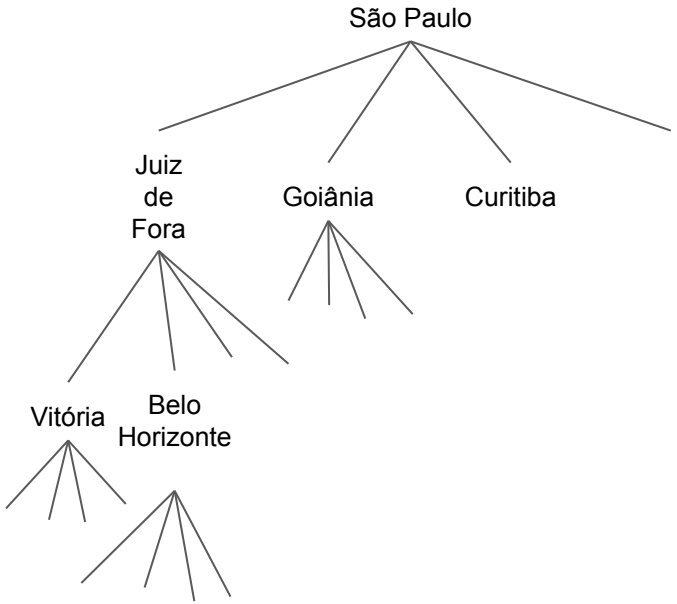
Inserção: Vitória

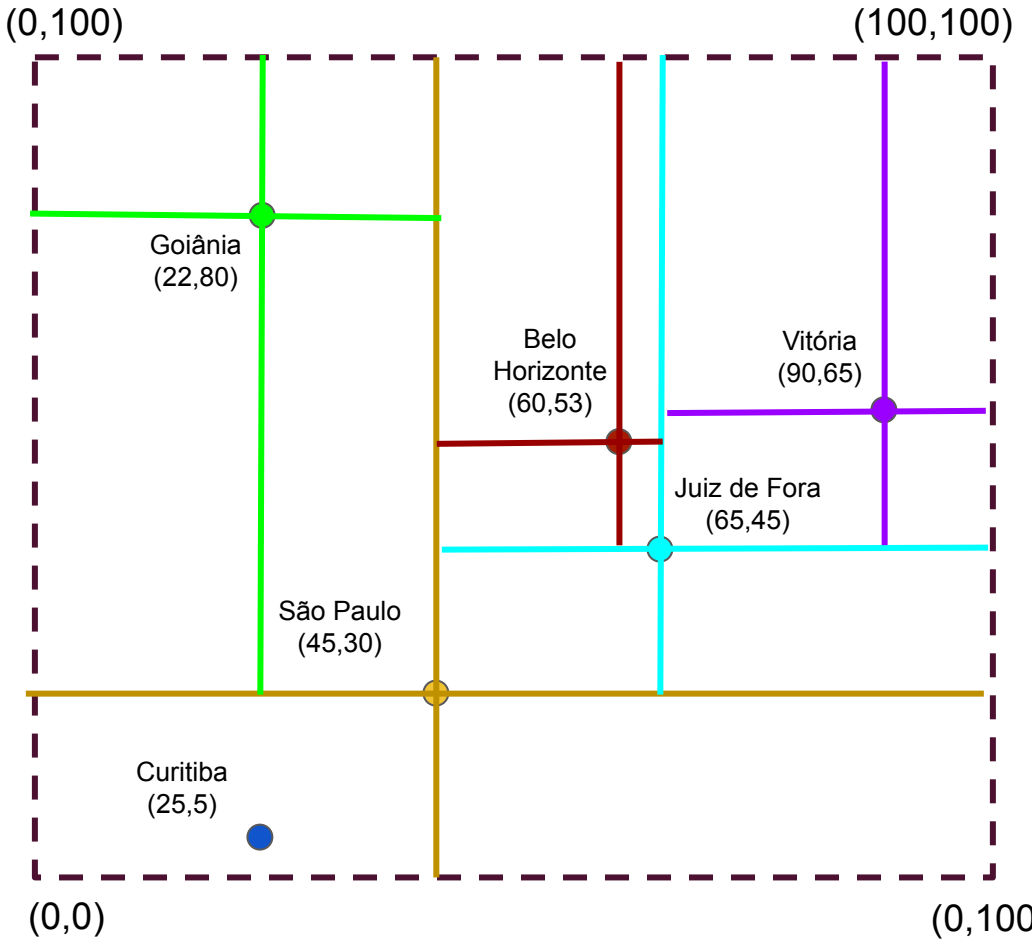




Quadtree - Exemplo

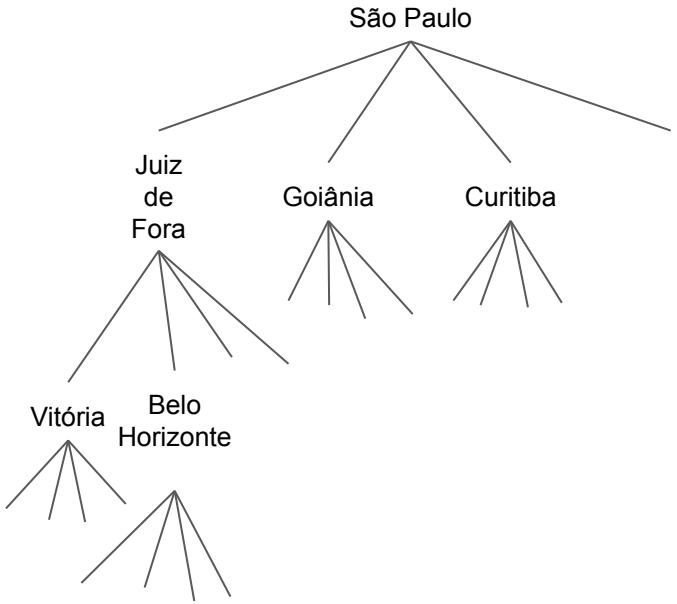
Inserção: Curitiba

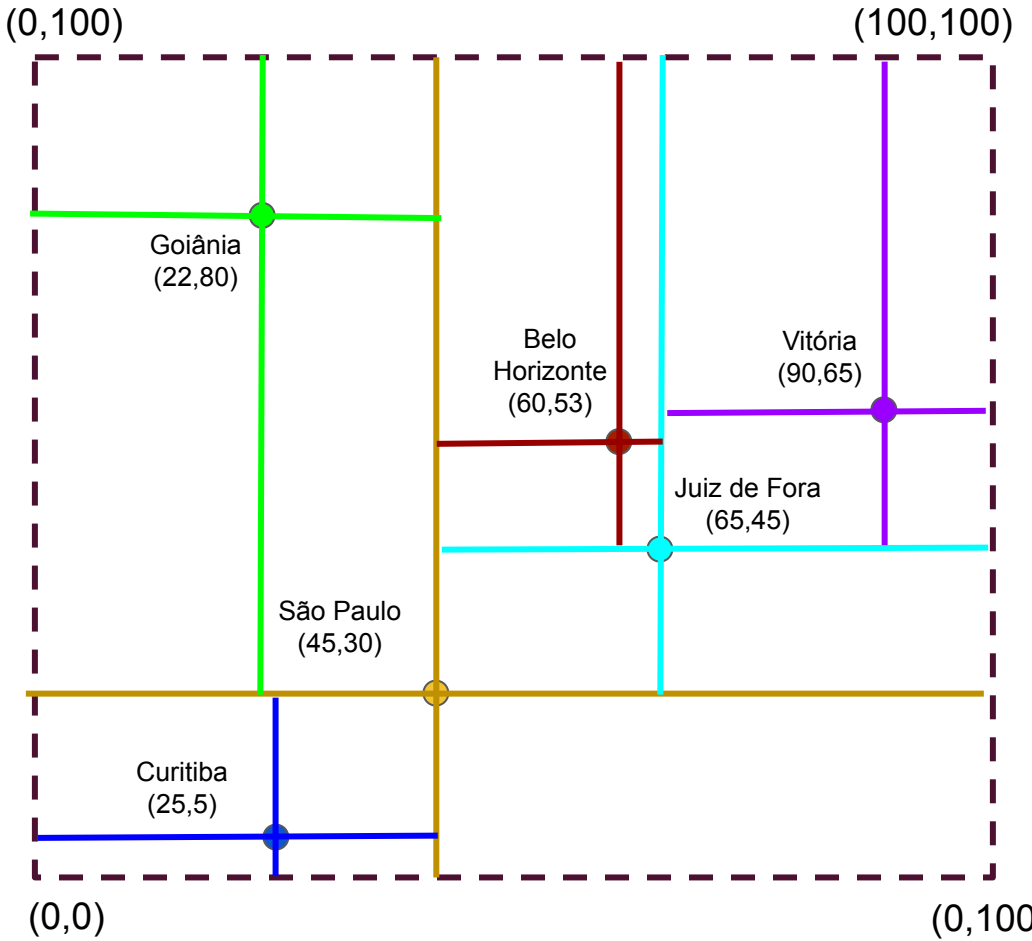




Quadtree - Exemplo

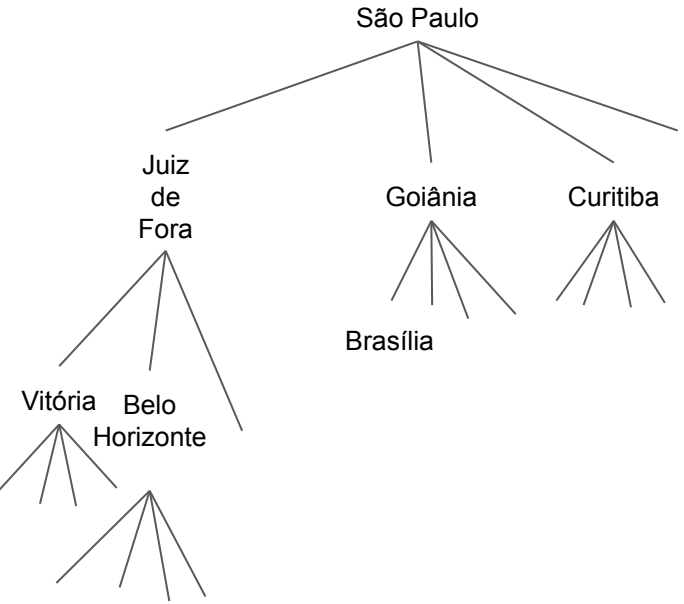
Inserção: Curitiba

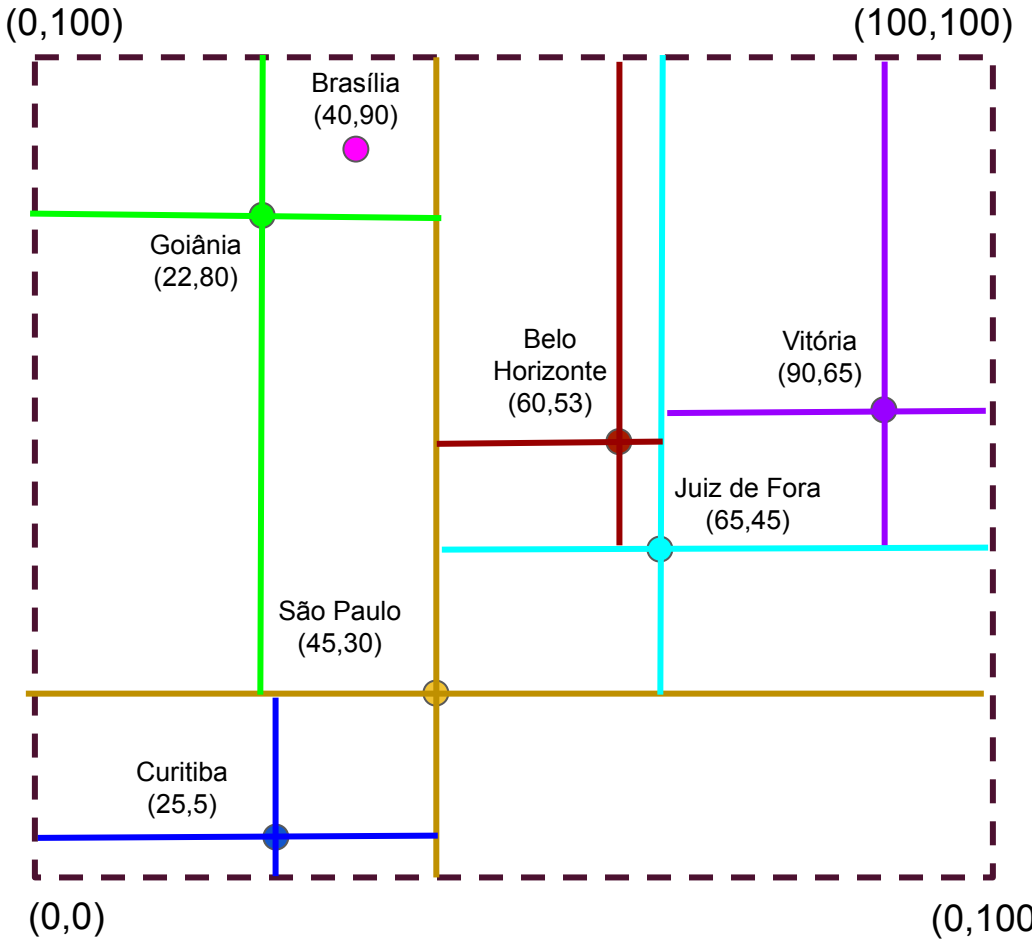




Quadtree - Exemplo

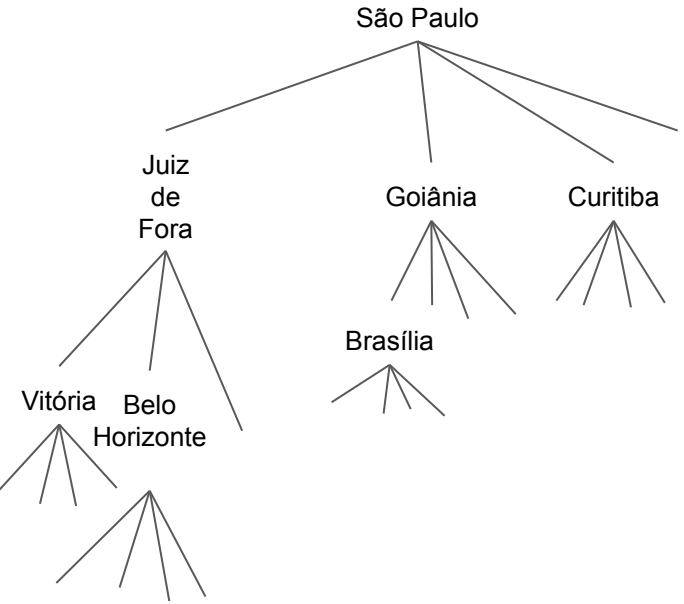
Inserção: Brasília

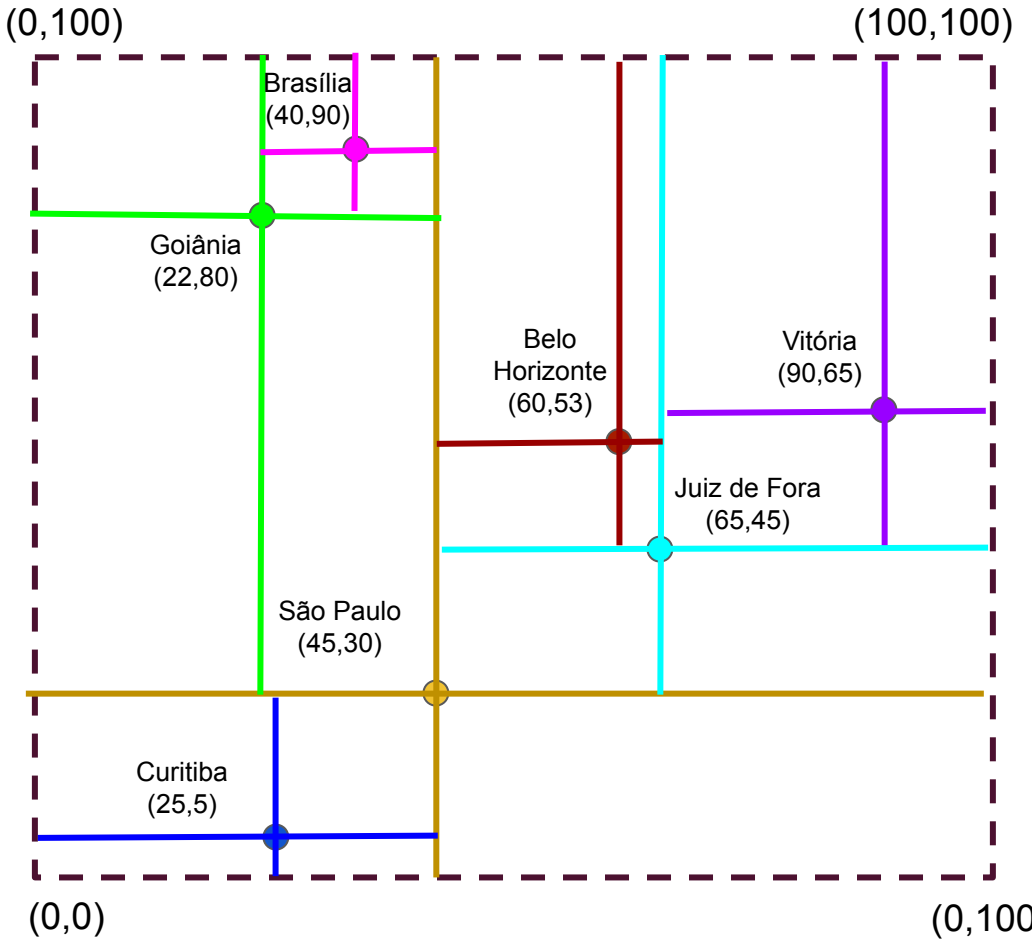




Quadtree - Exemplo

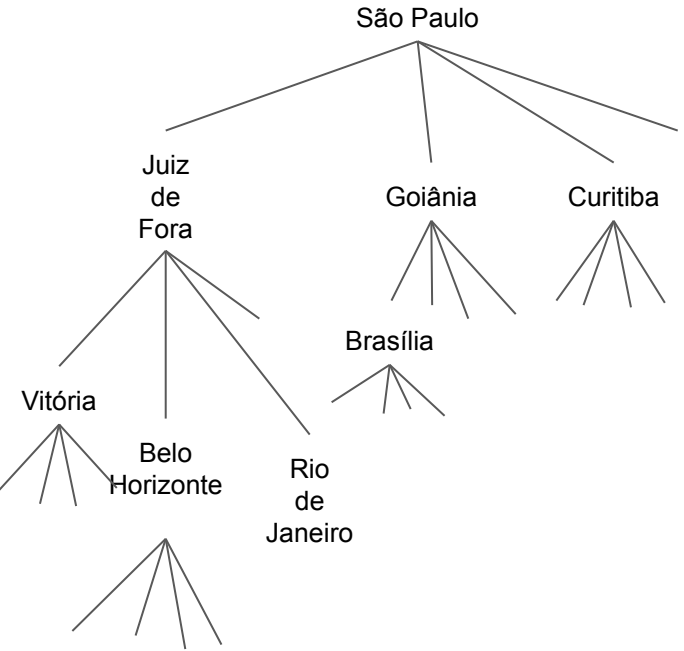
Inserção: Brasília

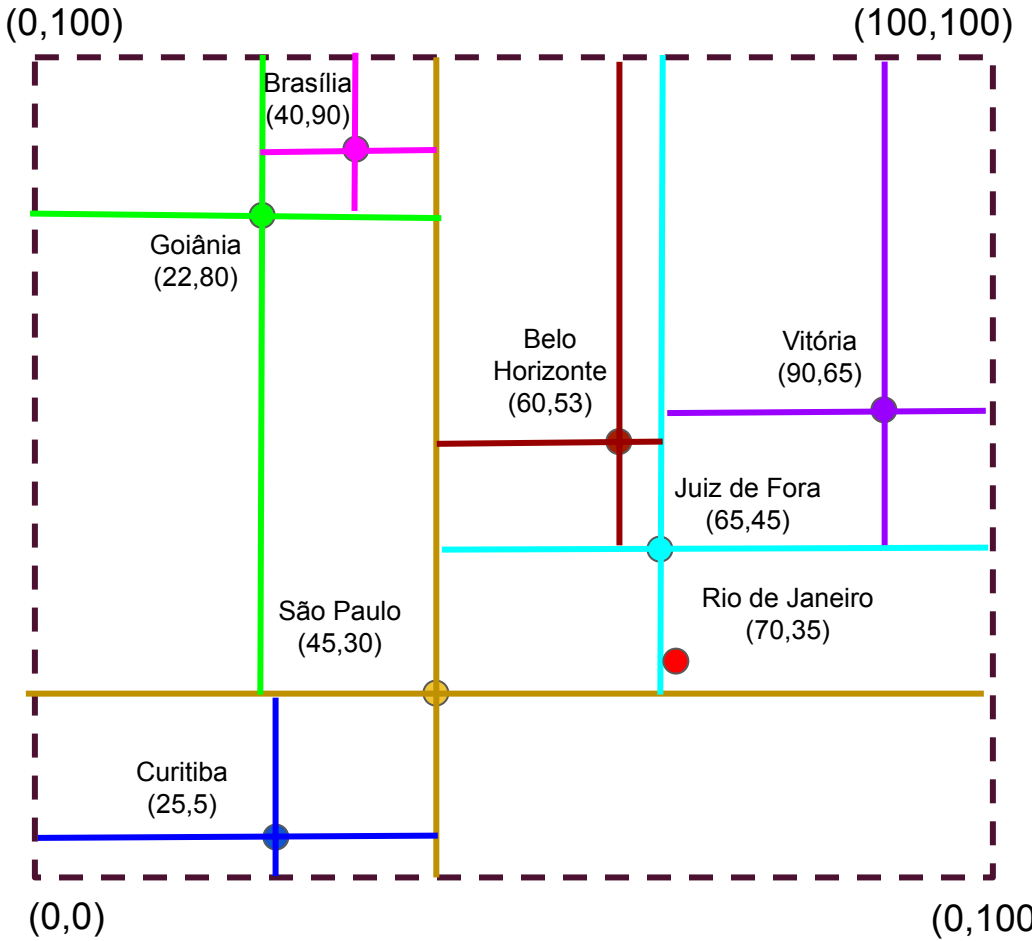




Quadtree - Exemplo

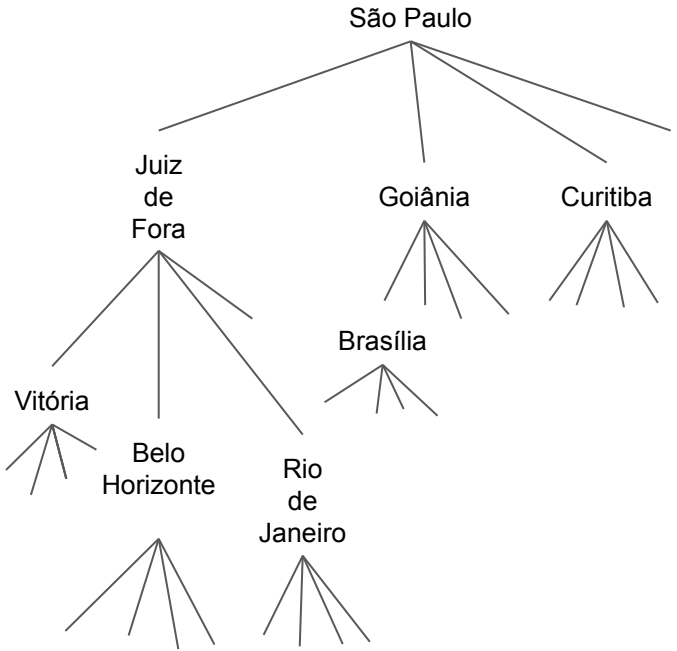
Inserção: Rio de Janeiro

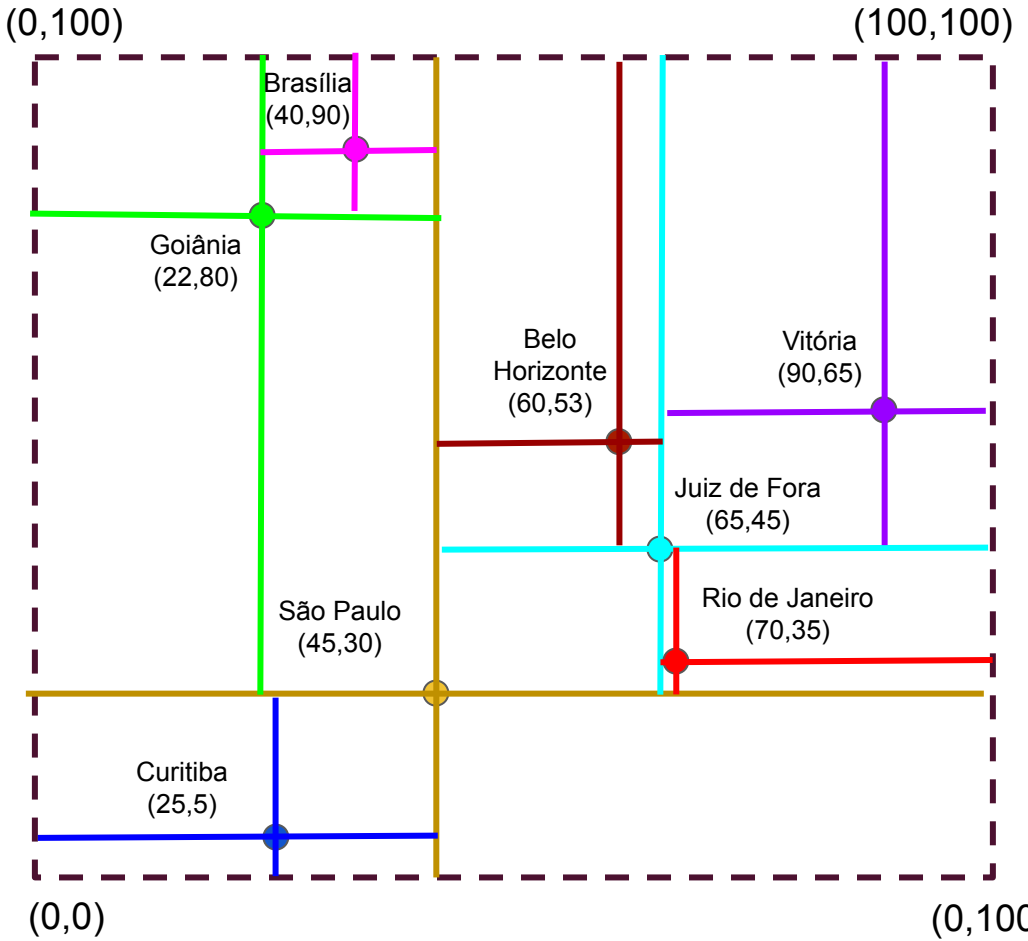




Quadtree - Exemplo

Inserção: Rio de Janeiro



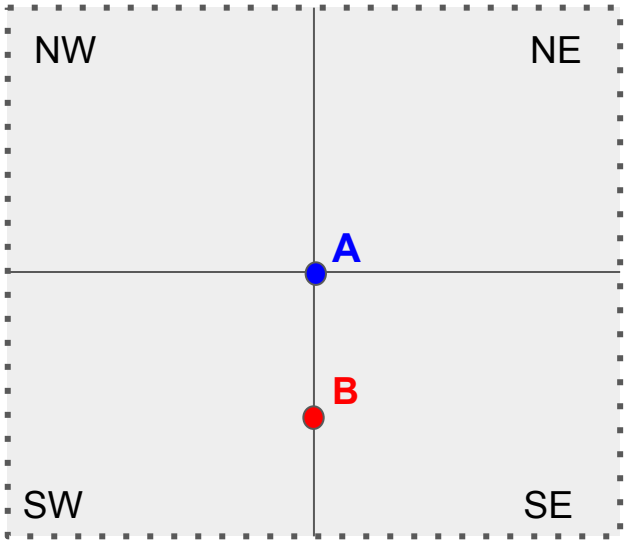


Importante!

- O que acontece quando pontos são inseridos nas arestas dos quadrantes?
- Regra:
  - Limites inferior e esquerdo de cada bloco são fechados
  - Limites superior e direito são abertos

Importante!!

- Pontos inseridos nas arestas dos quadrantes
- Regra:
  - Limites inferior e esquerdo de cada bloco são fechados
  - Limites superior e direito são abertos
- No exemplo ao lado:
  - B está a sudeste (SE) de A e não sudoeste (SW).



Algoritmo de Comparação:

```
Q = quadtree_compara(R, P)
   retorna o quadrante Q onde P está localizado no quadrante enraizado
   por R

   SE ( COORDX(P) < COORDX(R) ) ENTÃO
        SE (COORDY(P) < COORDY(R) ENTÃO
             retorna SW
        CASO CONTRÁRIO
             retorna NW
        FIM-SE
   CASO CONTRÁRIO
        SE (COORDY(P) < COORDY(R) ENTÃO
             retorna SE
        CASO CONTRÁRIO
             retorna NE
        FIM-SE
  FIM-SE
```

Algoritmo de Inserção

```
quadtree_insere(R, P)
   Insere P em uma quadtree enraizado por R
    SE (R == NIL) ENTÃO
        R = P
    CASO CONTRÁRIO
    ENQUANTO (R != NIL) E COORD(P) != COORD(R) FAÇA
        PAI = R
        Q = quadtree_compare(R,P)
        R = quadrante(R,Q)
    FIM-ENQUANTO
    SE (R == NIL) ENTÃO
        quadrante(PAI,Q) = P
    FIM-SE
FIM
```

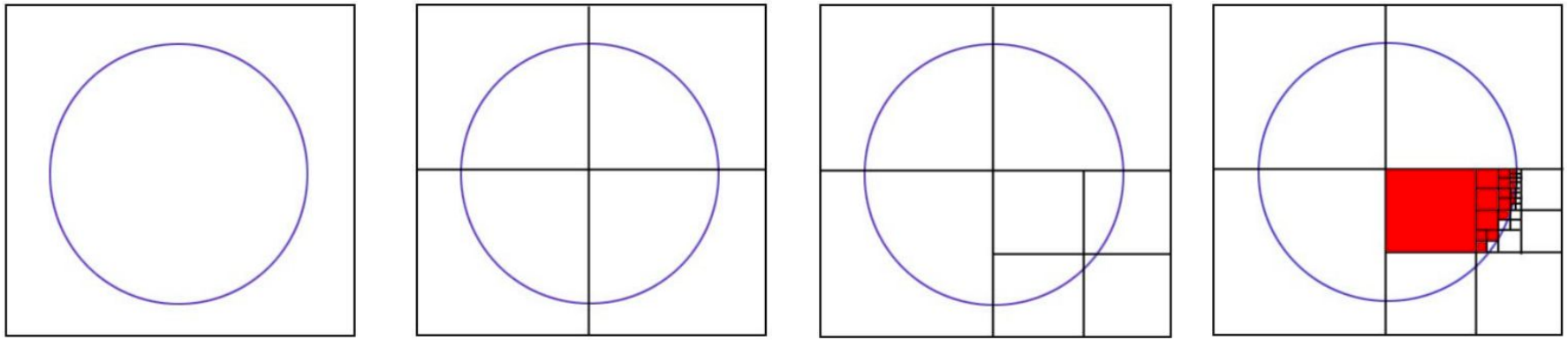
quadrante retorna o ponteiro da subárvore Q da árvore enraizada por R

Observações:

- A operação de remoção é muito complexa
- Solução:
  - Não remover os elementos mas apenas marcá-los como removidos.
  - De tempos em tempos, reconstruir a árvore excluindo os elementos marcados como removidos.

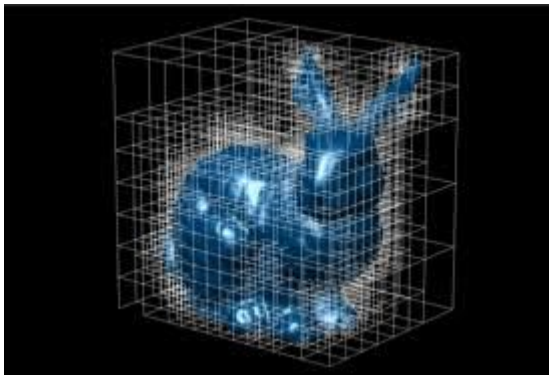
Outras Aplicações:

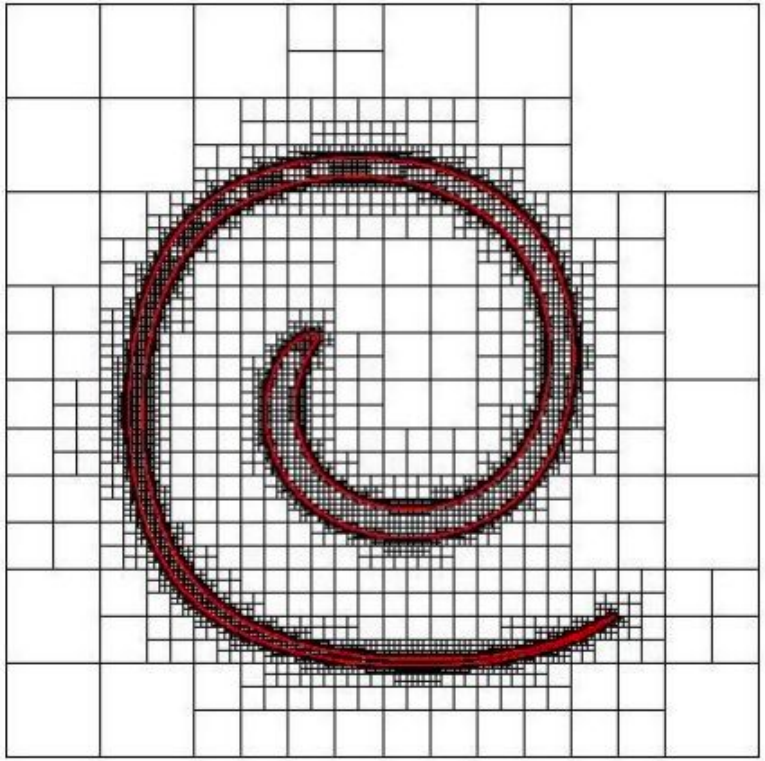
- Computação Gráfica: Preenchimento de Polígonos
  - Subdividir o espaço
  - Se o quadrante for homogêneo ou se o nível de detalhe foi alcançado, interrompe a divisão. Caso contrário, continue dividindo.



Outras Aplicações:

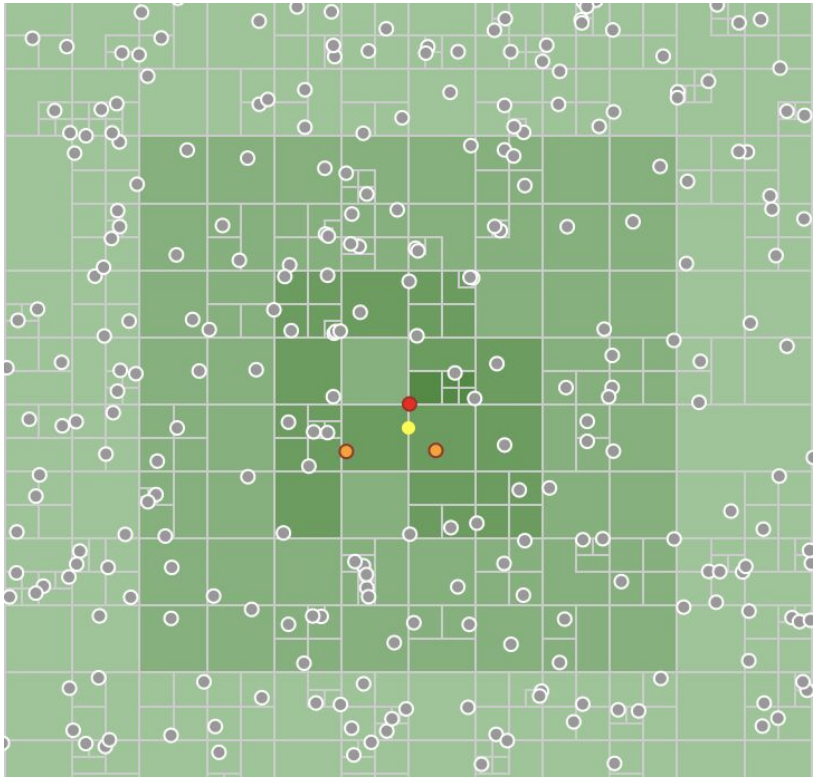
- Computação Numérica:
  - Geração de discretizações
  - refinamento adaptativo de malhas
- Em 3D: Octrees





Outras aplicações

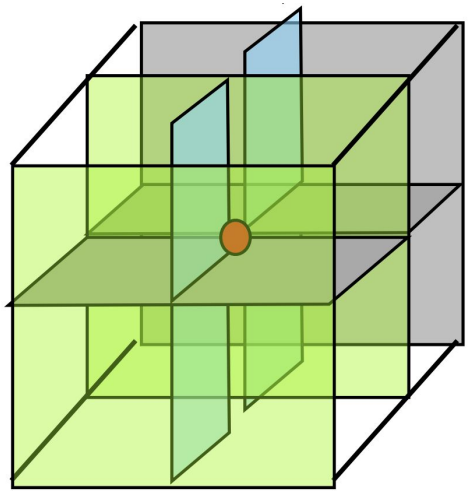
Localizar o vizinho mais próximo



http://bl.ocks.org/patricksurry/6478178

Quadtree

- Em resumo, é uma forma de dividir o espaço tal que seja simples caminhar e buscar elementos.
- Pode ser ineficiente e ocupar muito espaço ao generalizar para 3D
  - Octree - cada nó tem 8 ponteiros
  - em d dimensões cada nó tem 2^d ponteiros



- Existem variações
  - Baseadas em Trie: MX Quadtree, PR Quadtree
  - Baseadas em pontos: Point Quadtree
  - Octrees lineares

Exercícios:

Mostre a QuadTree resultante após a inserção dos pontos:

- (A, 51,30)
- (B, 13,70)
- (C, 81,40)
- (D, 81,70)
- (E, 02,25)
- (F, 01,01)
- (G, 99,99)
- (H, 63,30)
- (I, 70,67)
- (J, 50,50)

Referências:

- de Berg (M.T.), Marc Van Kreveld, Mark Overmars, Otfried Schwarzkopf, Computational Geometry: Algorithms and Applications, Springer Science & Business Media, 2000 - Computers - 367 pages.

- Slides do Prof. Jairo Francisco de Souza
  - https://www.ufjf.br/jairo_souza/ensino/material/ed2/
- Slides do Prof. Carl Kingsford
  - https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/